

# Almost overlap-free words and the word problem for the free Burnside semigroup satisfying $x^2 = x^3$

Andrew N. Plyushchenko, Arseny M. Shur

## Abstract

In this paper we investigate the word problem of the free Burnside semigroup satisfying  $x^2 = x^3$  and having two generators. Elements of this semigroup are classes of equivalent words. A natural way to solve the word problem is to select a unique “canonical” representative for each equivalence class. We prove that overlap-free words and so-called almost overlap-free words (this notion is some generalization of the notion of overlap-free words) can serve as canonical representatives for corresponding equivalence classes. We show that such a word in a given class, if any, can be efficiently found. As a result, we construct a linear-time algorithm that partially solves the word problem for the semigroup under consideration.

## Introduction

The free Burnside semigroups satisfying  $x^n = x^{n+m}$  are defined by the identities, which impose the equivalence of  $n$ -th and  $(n+m)$ -th powers of words in any context. Thus elements of these semigroups are classes of equivalent words. The structure of free Burnside semigroups is far from being completely described. However, a considerable progress was achieved in the 1990’s. Kaĉourek and Polák [6], de Luca and Varricchio [10], McCammond [11], Guba [4, 5], and do Lago [7, 8] produced a series of papers that led to the discovery of many structural properties of free Burnside semigroups. The reader is referred to the survey [9] for the history and the formulations of remarkable results.

The word problem is probably the most important and challenging combinatorial problem related to free Burnside semigroups. It is formulated as follows: *given words  $U$  and  $V$ , decide whether or not  $U$  and  $V$  are equivalent in a given semigroup.* In a series of papers [4, 5, 7, 10, 11] the word problem was solved for all free Burnside semigroups satisfying  $x^n = x^{n+m}$  for  $n \geq 3$  and  $m \geq 1$ . In this case, all equivalence classes are regular languages, and the deciding algorithm for the word problem constructs an automaton recognizing the class of the word  $U$  and tries to accept  $V$  by this automaton

(see [4, 5]). Due to Green and Rees [3] and later Kaĉourek and Polák [6], the word problem for the case  $n = 1$  was solved modulo periodic groups (i. e., reduced to the word problem for the groups satisfying  $x^m = 1$ ). For the case  $n = 2$ , this problem remains open (and some equivalence classes are not regular languages, see [8]). Note that the word problem for the particular case  $n = 2$  and  $m = 1$  was explicitly formulated by Brzozowski [2]. This case was considered to be the hardest one to analyze (see [7]). In what follows, we consider the free Burnside semigroup satisfying  $x^2 = x^3$  and having two generators.

A natural way to solve the word problem is to select a unique “canonical” representative for each equivalence class. Thus every word is equivalent to exactly one “canonical” word. If the latter can be efficiently found, then the word problem is decidable. It is clear that the choice of the canonical representatives is not an easy task. For example, we cannot take just a cube-free word as a representative, since there exist equivalent cube-free words. And if we take the shortest word in the class as a representative, it may be very hard to determine such a word.

As was proved in [13], overlap-free words can serve as canonical representatives for corresponding equivalence classes. In this paper we generalize this result for “almost” overlap-free words and show that such a word in a given class, if any, can be efficiently found. Thus we construct an efficient (in fact, linear-time) algorithm that partially solves the word problem for the semigroup under consideration.

To give precise formulations of the main results, we say a few words about definitions and notation.

Let  $\Sigma = \{a, b\}$ . As usual, we write  $\Sigma^*$  for the monoid of all words over  $\Sigma$  (including the empty word  $\lambda$ ) and  $\Sigma^+$  for the semigroup of all non-empty words over  $\Sigma$ . For a word  $W$ , its length is denoted by  $|W|$  and its  $i$ -th letter is denoted by  $W[i]$ ; thus,  $W = W[1] \dots W[|W|]$ . In the sequel, we write  $W[i \dots j]$  instead of  $W[i] \dots W[j]$ . *Factors*, *prefixes*, *suffixes*, and *powers* of a word are defined in the usual way. Recall that the *Kleene star*  $W^*$  of  $W$  is the union of all nonnegative powers of the word  $W$ .

A factor, prefix, or suffix of a word  $W$  is called *proper* if it is not equal to  $W$ . A factor of a word  $W$  is called *internal* if it is neither a prefix nor a suffix of  $W$ . We write  $U \leq V$  ( $U < V$ ,  $U \ll V$ ) if a word  $U$  is a factor (resp., proper factor, internal factor) of a word  $V$ .

A word  $U$  is called *overlap-free* if it contains no factor of the form  $XYXYX$  for any  $X \in \Sigma^+$ ,  $Y \in \Sigma^*$ . If  $U$  contains no **proper** factor of the form above, then we call it *almost overlap-free*. Finally, we call a word *cube-free* if it contains no factor of the form  $XXX$  for any  $X \in \Sigma^+$ .

Two words  $U$  and  $V$  are *neighbours* if and only if one of them can be obtained from another one by replacing some factor of the form  $Y^2$  by the factor  $Y^3$ . Thus we have the *neighbourhood relation*

$$\pi = \{(XY^2Z, XY^3Z), (XY^3Z, XY^2Z) \mid X, Z \in \Sigma^*, Y \in \Sigma^+\}.$$

The *free Burnside semigroup satisfying  $x^2 = x^3$  generated by  $\Sigma$*  is defined as the quotient semigroup  $\Sigma^+/\sim$ , where  $\sim$  is the smallest congruence containing  $\pi$ . If  $U \sim V$ , then the words  $U$  and  $V$  are said to be *equivalent*. The congruence class of  $U$  is denoted by  $[U]$ .

In these terms, the first result of this paper is formulated as follows.

**Theorem 1.** *Except for the classes  $[aa]$  and  $[bb]$ , each class of the congruence  $\sim$  contains at most one almost overlap-free word. Each of the two exceptional classes contains exactly two almost overlap-free words.*

Theorem 1 for overlap-free words was proved in [13]. In this paper the proof from [13] is simplified and applied to a more general case.

The second result of this paper now follows.

**Theorem 2.** *If at least one of words  $U$  and  $V$  is equivalent to an almost overlap-free word, then the word problem for the pair  $(U, V)$  can be solved in time  $O(n)$ , where  $n = \max\{|U|, |V|\}$ .*

In fact, we will construct Algorithm EqAOF (abbr. Equivalent Almost Overlap-Free), which returns the almost overlap-free word  $V$  that is equivalent to a given input word  $U$  or reports that no such word  $V$  exists. It should be mentioned that some equivalence classes contain no almost overlap-free words like the class  $[ababaa] = (ab)^*ababaa$ .

Sketches of the proofs of Theorems 1 and 2 were given in [12]. Here we present a full version of these proofs.

The text is subdivided into six sections. In Sect. 1 we introduce the main tools and techniques. Sect. 2 contains the proof of Theorem 1. Last four sections are devoted to the construction and analysis of Algorithm EqAOF.

## 1 The main tools and techniques

Recall that *Thue-Morse morphism*  $\varphi$  of  $\Sigma^+$  is defined by the rule

$$\varphi(a) = ab, \quad \varphi(b) = ba.$$

A  $\varphi$ -image is any word  $U \in \varphi(\Sigma^+)$ .

One can easily check

**Observation 1.1.** *The set  $\varphi(\Sigma^+)$  consists exactly of all even-length words such that all factors  $aa$  or  $bb$  start at even positions.*

The main idea of our solution to the word problem for an instance  $(U, V)$  is to simplify and shorten the words  $U$  and  $V$ . Each simplification is either an equivalent transformation of a word or a simultaneous transformation of a pair of words, preserving their equivalence/non-equivalence. The main instrument is the function  $\varphi^{-1}$  applied to a pair of words. This function reduces the length of both words to one half. All other transformations are needed to get a pair  $(U', V')$  of  $\varphi$ -images  $U'$  and  $V'$ . These transformations are

- complete reduction of a word, which is an equivalent transformation on the class of so-called  $\widetilde{AB}$ -whole words;
- tail reductions applied to a pair of words in order to make the words  $\widetilde{AB}$ -whole;
- functions  $\xi$  and  $\eta$  applied to a pair of completely reduced words in order to turn them into  $\varphi$ -images.

In this preliminary section we introduce all these transformations and study their properties.

**1.1 Uniformity and complete reduction.** Following [14], we generalize the notion of  $\varphi$ -image. Namely we say that a word  $U$  is *uniform* if all its factors  $aa$  or  $bb$  start in  $U$  either always at even positions or always at odd positions. Otherwise a word is called *non-uniform*. We call a word *letter-alternating* if it contains no factor  $aa$  or  $bb$ . In what follows,  $\tilde{A}$  (resp.,  $\tilde{B}$ ) abbreviates an arbitrary letter-alternating word of the form  $aba(ba)^*$  (resp.,  $bab(ab)^*$ ). All letter-alternating words are obviously uniform.

Uniformity plays a crucial role in subsequent considerations. First, uniform words form a vast majority among all almost overlap-free words. Second, an important connection between the uniformity, Thue-Morse morphism, and the congruence  $\sim$  was established in [1]:

**Proposition 1.1.** *Suppose  $W \sim \varphi(U)$  for some  $U \in \Sigma^+$  and a uniform word  $W$ . Then there exists a word  $V \in \Sigma^+$  such that  $W = \varphi(V)$  and  $V \sim U$ .*

According to Proposition 1.1, since Thue-Morse morphism preserves the congruence  $\sim$ , we have  $\varphi(V) \sim \varphi(U)$  if and only if  $V \sim U$ . Thus the word problem for two  $\varphi$ -images  $U$  and  $V$  can be reduced to the word problem for the pair of shorter words  $\varphi^{-1}(U)$  and  $\varphi^{-1}(V)$ . So, we are going to replace considered words by  $\varphi$ -images whenever it is possible. First, we introduce three reduction operations in order to transform any given word to a uniform word (see [1]).

The first operation is called  *$r_1$ -reduction*. It reduces all factors of the form  $c^n$  to  $c^2$ , where  $c \in \Sigma$  and  $n > 2$ . The result of this operation applied to a word  $U$  is denoted by  $r_1(U)$ . We say that a word  $U$  is  *$r_1$ -reduced* if  $U = r_1(U)$ .

Obviously,  $U \sim r_1(U)$  for any  $U \in \Sigma^+$ . Moreover,  $r_1$ -reduction preserves the relation  $\pi$  as well:  $(U, V) \in \pi$  implies  $(r_1(U), r_1(V)) \in \pi$  or  $r_1(U) = r_1(V)$  (see [1]). Thus if we denote by  $\pi_{r_1}$  and  $\sim_{r_1}$  the restrictions of relations  $\pi$  and  $\sim$  respectively to the set of all  $r_1$ -reduced words, then we get  $\sim_{r_1} = \pi_{r_1}^+$ . Therefore it is sufficient to solve the word problem for  $r_1$ -reduced words only. For this reason, in the sequel we usually consider  $r_1$ -reduced words.

Let us denote the class of all  $r_1$ -reduced words that are equivalent to  $U$  by  $[U]_{r_1}$ . Obviously, all almost overlap-free words, except for the words  $aaa$  and  $bbb$ , are  $r_1$ -reduced. Hence, Theorem 1 can be reformulated as follows:

(\*) *for any word  $U$ , the class  $[U]_{r_1}$  contains at most one almost overlap-free word.*

Actually we will prove Theorem 1 in this form.

Now let  $U$  be an  $r_1$ -reduced word. Then  $r_A(U)$  is the word obtained from  $U$  by performing all possible reductions of the form

$$a\tilde{A}a \rightarrow aa. \quad (1.1a)$$

The word  $r_B(U)$  is defined in a symmetric way using reductions of the form

$$b\tilde{B}b \rightarrow bb. \quad (1.1b)$$

Finally, let  $r(U) = r_B(r_A(r_1(U)))$  for an arbitrary word  $U \in \Sigma^*$ . The operation  $r$  is called *complete reduction*. We call a word  $U$  *completely reduced*

if  $r(U) = U$ . As it was shown in [1], the word  $r(U)$  can be obtained from  $r_1(U)$  by performing all possible reductions of the form (1.1a) and (1.1b) **in any order**.

It is easy to check that, in contrast to  $r_1$ -reductions, some complete reductions do not preserve the congruence  $\sim$ . However, as we will see later, under certain conditions even the complete reduction preserves  $\sim$ .

It appears that completely reduced words are exactly uniform ones:

**Proposition 1.2.** *The following three conditions are equivalent for an arbitrary word  $U$ :*

- (1)  $U$  is a factor of a  $\varphi$ -image;
- (2)  $U$  is uniform;
- (3)  $U$  is completely reduced.

The equivalence of (2) and (3) was proved in [13], while the equivalence of (1) and (2) follows from definitions.

**1.2 Reduction of  $\widetilde{AB}$ -whole words. Non-reducible tails.** So, we can reduce any word  $U$  to the uniform word  $r(U)$ . Proposition 1.4 below states that words  $U$  and  $r(U)$  are equivalent under certain conditions. This proposition uses the important notions of  $\widetilde{A}$ -whole and  $\widetilde{B}$ -whole words (see [1]). An  $r_1$ -reduced word  $W$  is called  $\widetilde{A}$ -whole if every factor  $X$  of the form  $a\widetilde{A}a$  occurs in  $W$  inside the factor  $abXba$ . The notion of  $\widetilde{B}$ -whole word is dual to the above one. If a word is  $\widetilde{A}$ -whole and  $\widetilde{B}$ -whole, it will be called  $\widetilde{AB}$ -whole. Obviously, any completely reduced word is  $\widetilde{AB}$ -whole.

In the sequel we often use the following proposition proved in [1].

**Proposition 1.3.** *If  $r_1$ -reduced words  $U$  and  $V$  are equivalent and  $U$  is  $\widetilde{A}$ -whole ( $\widetilde{B}$ -whole), then  $V$  is  $\widetilde{A}$ -whole (resp.,  $\widetilde{B}$ -whole) as well.*

Now we ready to formulate Proposition 1.4.

**Proposition 1.4.** *Let  $U$  be an  $\widetilde{AB}$ -whole word. Then  $U \sim r(U)$  if and only if  $U$  has no prefix of the form  $(aba)(aba)^*(ab)^2(ab)^*aa$  and no suffix of the form  $aa(ba)^*(ba)^2(aba)^*(aba)$  up to negation.*

We refer to the prefixes and suffixes mentioned in Proposition 1.4 as *non-reducible tails* and distinguish four kinds of such tails according to the following table:

	A-tail	B-tail
left (prefix)	$(aba)(aba)^*(ab)^2(ab)^*aa$	$(bab)(bab)^*(ba)^2(ba)^*bb$
right (suffix)	$aa(ba)^*(ba)^2(aba)^*(aba)$	$bb(ab)^*(ab)^2(bab)^*(bab)$

In order to prove Proposition 1.4, we need an auxiliary result.

**Lemma 1.1.** *Let  $U$  and  $V$  be  $r_1$ -reduced words,  $U \sim V$ , and let  $U$  have a non-reducible tail. Then the word  $V$  has a non-reducible tail of the same kind.*

**Proof.** It is sufficient to prove the statement of the lemma for a pair of neighbours. Let  $U = XY^kZ$ ,  $V = XY^lZ$ , where  $X, Z \in \Sigma^*$ ,  $Y \in \Sigma^+$ , and  $\{k, l\} = \{2, 3\}$ . Without loss of generality assume that  $U$  has a left A-tail, that is,  $U = (aba)^n(ab)^m aaT$ , where  $n \geq 1$ ,  $m \geq 2$ , and  $T \in \Sigma^*$ . The words

$U$  and  $V$  have the common prefix  $XY Y$ , therefore the proof is evident if  $(aba)^n(ab)^m aa$  is a prefix of the word  $XY Y$ . So suppose that  $XY Y$  is a proper prefix of the word  $(aba)^n(ab)^m aa$ . Consider all possible cases.

First suppose that  $|X| \geq |(aba)^n|$ . In this case,  $YY \leq (ab)^m a$ . Hence  $Y$  is a letter-alternating word of even length. Since  $Y^k$  is also letter-alternating, we have  $Y^k \leq (ab)^m a$ . Then the word  $V$  has the prefix  $(aba)^n Sa$  for some neighbour  $S$  of the word  $(ab)^m a$ . Thus  $V$  has a left  $A$ -tail, as desired.

Now let  $|X| < |(aba)^n|$  and  $|XY| > |(aba)^n|$ . Then  $Y$  contains the factor  $aa$  obtained from the last letter of  $(aba)^n$  and the first letter of  $(ab)^m a$ . On the other hand, the suffix  $Y$  of  $XY Y$  is letter-alternating as a factor of the word  $(ab)^m a$ . We get a contradiction. Hence, this case is impossible.

Finally, let  $|XY| \leq |(aba)^n|$ . Then the word  $Y$  contains no factors  $abab$  and  $baba$ . Thus,  $XY^2 \leq (aba)^n(aba)$  whence the word  $Y^2$  also contains no factors  $abab$  and  $baba$ , in particular,  $Y \notin \{ab, ba\}$ . Since the words  $U$  and  $V$  are  $r_1$ -reduced, we get  $Y \notin \{a, b, aa, bb\}$ . Therefore,  $|Y| > 2$  and the factors  $abab$  and  $baba$  can occur in  $Y^k$  inside the factor  $Y^2$  only. Hence the word  $Y^k$  contains no factors  $abab$  and  $baba$  as well, and we have  $XY^k \leq (aba)^{n+1}$ . So the word  $V$  has the prefix  $Sb(ab)^{m-2}aa$  for some neighbour  $S$  of the word  $(aba)^{n+1}$ . One can easily check that the prefix  $Sb(ab)^{m-2}aa$  is a left  $A$ -tail. This completes the proof of the lemma.  $\square$

**Proof of Proposition 1.4.** Let a word  $U$  be  $\widetilde{AB}$ -whole and have no non-reducible tails. Recall that reductions of kind (1.1a) and (1.1b) can be applied in any order. So, to prove the forward implication it is sufficient to find a sequence of reductions from  $U$  to  $r(U)$  such that every single reduction preserves the relation  $\sim$ . Indeed, by Proposition 1.3, the word obtained by a single reduction will remain  $\widetilde{AB}$ -whole and, by Lemma 1.1, this word will have no non-reducible tails.

We will reduce  $U$  as follows: first perform all possible reductions of the form  $aabaa \rightarrow aa$  and  $bbabb \rightarrow bb$  (in any order) and then all other reductions (also, in any order).

Suppose that  $U = Xaaba aY$ , and  $U' = XaaY$  is obtained from  $U$  by the reduction  $aabaa \rightarrow aa$ . Since  $U$  is  $\widetilde{A}$ -whole, we have  $U = X'abaaba aY'$ , where  $X = X'ab$  and  $Y = baY'$ . Thus we get  $U = X'(aba)^3Y'$ ,  $U' = X'(aba)^2Y'$  whence  $U \sim U'$ . A single reduction  $bbabb \rightarrow bb$  is examined symmetrically.

Now suppose that  $U$  contains no factors  $aabaa$  and  $bbabb$ . Without loss of generality assume that  $U = Xa(ab)^k aaY$ , where  $k \geq 2$ , and let  $U'$  be obtained from  $U$  by the reduction  $a(ab)^k aa \rightarrow aa$ , that is,  $U' = XaaY$ . Since  $U$  is an  $\widetilde{A}$ -whole word, we can write  $U = X'aba(ab)^k aabaY'$ , where  $X = X'ab$  and  $Y = baY'$ . If  $X' = \lambda$  or  $Y' = \lambda$ , then  $U$  has a non-reducible left (resp., right) tail, which is impossible by the conditions of the proposition. Hence,  $X' \neq \lambda$  and  $Y' \neq \lambda$ . Since the word  $U$  contains no factors  $aabaa$  and  $bbabb$ , we have  $U = X''baba(ab)^k aababY''$ . Then

$$\begin{aligned} U &= X''(baba)(ab)^k a(abab)Y'' \sim X''(ba)^{k+1}(ab)^k a(abab)^{k+1}Y'' = \\ &X''b(ab)^k a(ab)^k a(ab)^k a bY'' \sim X''b(ab)^k a(ab)^k abY'' = \\ &X''(ba)^{k+1}(ab)^{k+1}Y'' \sim X''babaababY'' = U'. \end{aligned}$$

We get  $U \sim U'$ , as desired. So, the forward implication is proved.

The backward implication ( $r(U) \sim U$ ) implies that  $U$  has no non-reducible tails) trivially follows from Lemma 1.1, since  $r(U)$  has no non-reducible tails.  $\square$

Our prime interest is in the study of equivalence classes that contain almost overlap-free words. Since almost overlap-free words have no non-reducible tails, all words from such equivalence classes have no non-reducible tails by Lemma 1.1. So, if a word  $U$  is  $\widetilde{AB}$ -whole, we can replace  $U$  by the word  $r(U)$ , which is equivalent to  $U$  by Proposition 1.4, and thus significantly simplify further analysis. The case when a word  $U$  is not  $\widetilde{AB}$ -whole will be considered below in Subsection 1.4.

**1.3 Uniform neighbours and quasi-neighbours.** This subsection is devoted to the study of uniform neighbours. We already know that  $\sim_{r_1} = \pi_{r_1}^+$ . Let us denote the restrictions of the relations  $\sim$  and  $\pi$  to the set of all uniform words by  $\sim_r$  and  $\pi_r$  respectively. In this subsection we prove the following equality:

**Proposition 1.5.**  $\sim_r = \pi_r^+$ .

First, we give more definitions. We say that words  $U$  and  $V$  are *ab-neighbours* and write  $(U, V) \in \pi_{ab}$  if one of them has the form  $X(ab)^2Z$  and the other has the form  $X(ab)^3Z$  for some  $X, Z \in \Sigma^*$ . Thus we get the *ab-neighbourhood* relation, which obviously preserves the uniformity. We call words  $U$  and  $V$  *quasi-neighbours* if  $U = V$  or there exist two sequences  $\{U_i\}_{i=1}^n$  and  $\{V_j\}_{j=1}^m$  of words such that  $U_1 = U$ ,  $(U_i, U_{i+1}) \in \pi_{ab}$  for each  $i = 1, \dots, n-1$ ;  $V_1 = V$ ,  $(V_j, V_{j+1}) \in \pi_{ab}$  for each  $j = 1, \dots, m-1$ ; and  $(U_n, V_m) \in \pi$ .

The proof of Proposition 1.5 is based on the following lemma.

**Lemma 1.2.** *Let  $\widetilde{AB}$ -whole words  $U$  and  $V$  have no non-reducible tails. If  $U$  and  $V$  are neighbours, then the words  $r(U)$  and  $r(V)$  are quasi-neighbours.*

**Proof.** Without loss of generality let  $U = XY Y Z$  and  $V = XY Y Y Z$  for some  $X, Z \in \Sigma^*$  and  $Y \in \Sigma^+$ . First, we apply the  $r$ -reduction to the words  $XY[1]$ ,  $Y$ , and  $Y[Y]Z$ . Obviously,  $r(XY[1]) = X'Y[1]$  and  $r(Y[Y]Z) = Y[Y]Z'$  for some  $X', Z' \in \Sigma^*$ , and  $r(Y)$  begins with  $Y[1]$  and ends by  $Y[Y]$ . Thus we obtain the neighbours  $X'r(Y)r(Y)Z'$  and  $X'r(Y)r(Y)r(Y)Z'$ . Clearly, these words are  $\widetilde{AB}$ -whole and have no non-reducible tails. So, to simplify notation we may assume that the words  $XY[1]$ ,  $Y$ , and  $Y[Y]Z$  are already  $r$ -reduced. Suppose that at least one of the words  $U$  and  $V$  is not  $r$ -reduced, and a word  $P$  is its factor of the form  $a\widetilde{A}a$  or  $b\widetilde{B}b$ . Without loss of generality we assume that  $P = a(ab)^k a a$  for some integer  $k > 0$ . The word  $P$  either contains  $Y$  as internal factor or occurs inside the factors  $XY$ ,  $YY$ , or  $YZ$ . Consider first the cases when  $Y \ll P$  or  $P \leq YY$ .

Suppose  $Y \ll P$ . Then the word  $Y$  is letter-alternating. If  $Y$  has odd length, then the word  $V = XY Y Y Z$  contains the factor  $Y[Y]YY[1]$  of the form  $a\widetilde{A}a$  or  $b\widetilde{B}b$ . The reduction of this factor turns  $V$  into  $U$ , and the lemma readily follows. If the length of  $Y$  is even, then the words  $YY$  and  $YYY$  are also letter-alternating. Hence the factor  $P$  begins inside  $X$  and ends inside  $Z$ , that is,  $P = X'Y^k Z'$ , where  $X'$  is a suffix of  $X$ ,  $Z'$  is a prefix of  $Z$ , and

either  $k = 2$  and  $P \leq U$  or  $k = 3$  and  $P \leq V$ . Clearly, the words  $X'YYZ'$  and  $X'YYYZ'$  both have the form  $a\tilde{A}a$ . Thus we reduce the factor  $X'YYZ'$  in  $U$  and the factor  $X'YYYZ'$  in  $V$  to get two identical words and prove the lemma.

In the case  $P \leq YY$ , we have  $P = aY_2Y_1a$ , where the words  $Y_1a$  and  $aY_2$  are respectively a prefix and a suffix of  $Y$  such that  $Y_2Y_1 = (ab)^k a$ . Obviously, the prefix  $Y_1$  of the word  $Y$  does not overlap the suffix  $Y_2$  of  $Y$  (otherwise  $Y_1$  contains the factor  $aY_2[1] = aa$ , which is impossible). Thus,  $Y = Y_1Y'Y_2$  for some  $Y' \in \Sigma^*$ . We have

$$U = XY_1Y'(Y_2Y_1)Y'Y_2Z \text{ and } V = XY_1Y'(Y_2Y_1)Y'(Y_2Y_1)Y'Y_2Z,$$

and after the reduction  $P \rightarrow aa$  we get the neighbours  $XY_1Y'Y'Y_2Z$  and  $XY_1Y'Y'Y'Y_2Z$ .

Now suppose that the word  $YY$  is already  $r$ -reduced and  $Y$  is not an internal factor of  $P$ . Then  $P$  occurs into both words  $U$  and  $V$  inside the factor  $XY$  or  $YZ$ . First, assume that  $YZ$  is  $r$ -reduced. Then  $P \leq XY$  and  $P$  is a unique factor of the form  $a\tilde{A}a$  or  $b\tilde{B}b$  in  $U$  and  $V$ . So,  $P = aX_1Y_1a$ , where  $aX_1$  is a suffix of  $X$ ,  $Y_1a$  is a prefix of  $Y$ ,  $X_1 \in \Sigma^*$ ,  $Y_1 \in \Sigma^+$ , and  $X_1Y_1 = (ab)^k a$ . Let  $Y = Y_1aY'$  for some  $Y' \in \Sigma^*$ . Since the words  $U$  and  $V$  are  $\widetilde{AB}$ -whole, we have  $X = X'abaX_1$  for some  $X' \in \Sigma^*$ . If  $k = 1$ , then

$$U = X'aba(aba)aY'YZ \text{ and } V = X'aba(aba)aY'YYZ.$$

After the reduction  $P \rightarrow aa$  we obtain the neighbours

$$X' \underbrace{aba}_{X_1Y_1} aY'YZ = X'X_1 \underbrace{Y_1aY'}_Y YZ = X'X_1YYZ$$

and

$$X' \underbrace{aba}_{X_1Y_1} aY'YYZ = X'X_1 \underbrace{Y_1aY'}_Y YYZ = X'X_1YYYZ.$$

In the case  $k \geq 2$ , we have  $X' \neq \lambda$  (if  $X' = \lambda$ , then both words  $U$  and  $V$  have the non-reducible tail  $(aba)(ab)^k aa$ , in contradiction with the lemma's condition). Since  $X$  is  $r$ -reduced, the last letter of  $X'$  cannot be equal to  $a$ , therefore  $X = X''babaX_1$ . After the reduction  $P \rightarrow aa$  we obtain the quasi-neighbours

$$U' = X''babaaY'YZ \text{ and } V' = X''babaaY'YYZ.$$

Indeed, if we put

$$U_i = X''ba(ba)^i aY'YZ \text{ and } V_i = X''ba(ba)^i aY'YYZ$$

for  $i = 1, \dots, k$ , we get  $U_1 = U'$ ,  $V_1 = V'$ ,  $(U_i, U_{i+1}) \in \pi_{ab}$ ,  $(V_i, V_{i+1}) \in \pi_{ab}$  for each  $i = 1, \dots, k-1$ , and the words

$$U_k = X''b \underbrace{a(ba)^k}_{X_1Y_1} aY'YZ = X''bX_1 \underbrace{Y_1aY'}_Y YZ = X''bX_1YYZ$$

and

$$V_k = X''b \underbrace{a(ba)^k}_{X_1Y_1} aY'YYZ = X''bX_1 \underbrace{Y_1aY'}_Y YYZ = X''bX_1YYYZ$$



are neighbours. So, since  $U' = r(U)$  and  $V' = r(V)$ , the words  $r(U)$  and  $r(V)$  are quasi-neighbours, as desired.

The case  $P \leq YZ$  is considered in the same way. Finally, if both words  $XY$  and  $YZ$  are not  $r$ -reduced, then there exist two words  $P_1$  and  $P_2$  of the form  $a\tilde{A}a$  or  $b\tilde{B}b$  such that  $P_1 \leq XY$  and  $P_2 \leq YZ$ . In this case, after the reductions of  $P_1$  and  $P_2$  we obtain the words  $r(U)$  and  $r(V)$ , which appear to be quasi-neighbours. Indeed, we can construct two sequences of words, instead of the one in the previous case, in the same way, increasing degree of the factor  $ab$  or  $ba$  in the prefix  $r(XY)$  and in the suffix  $r(YZ)$  of the words  $r(U)$  and  $r(V)$ . These sequences obviously satisfy all conditions from the definition of quasi-neighbours. This completes the proof.  $\square$

We should mention that a weaker version of Lemma 1.2 for the case when  $U$  and  $V$  are equivalent to some  $\varphi$ -images was proved in [1].

**Proof of Proposition 1.5.** We say that a sequence  $\{R_i\}_{i=0}^n$  of words is called a *linking*  $(U, V)$ -sequence if  $R_0 = U$ ,  $R_n = V$ , and  $(R_{i-1}, R_i) \in \pi$  for each  $k = 1, \dots, n$ . If all the words  $R_i$  are  $r_1$ -reduced ( $r$ -reduced), we call such a sequence an  $r_1$ -linking (resp., an  $r$ -linking)  $(U, V)$ -sequence. In these terms, Proposition 1.5 means that two uniform words are equivalent if and only if there exists an  $r$ -linking  $(U, V)$ -sequence.

If there exists an  $r$ -linking  $(U, V)$ -sequence, then  $U \sim V$  and both words  $U$  and  $V$  are uniform. Conversely, let  $U$  and  $V$  be uniform words and  $U \sim V$ . Then there exists an  $r_1$ -linking  $(U, V)$ -sequence  $\{W_k\}_{k=0}^n$ . Since the words  $U$  and  $V$  are uniform, all words  $W_k$  are  $\widetilde{AB}$ -whole and have no non-reducible tails. Consider the sequence  $\{W'_k = r(W_k)\}_{k=0}^n$ . We have  $W'_0 = U$ ,  $W'_n = V$ , and, by Lemma 1.2, the words  $W'_{k-1}$  and  $W'_k$  are uniform quasi-neighbours for each  $k = 1, \dots, n$ . So, the sequence  $\{W'_k\}_{k=0}^n$  is an  $r$ -linking  $(U, V)$ -sequence, as desired.  $\square$

#### 1.4 Non-uniform almost overlap-free words. Non-uniform tails.

All results mentioned in this subsection were proved in [13]. In the sequel, we often make use of the negation operation  $\overline{\phantom{x}}$ , which is a unique non-trivial automorphism of  $\Sigma^*$ . For example,  $\varphi(\overline{W}) = \overline{\varphi(W)}$ . We also write  $\overline{\mathcal{L}} = \{\overline{W} \mid W \in \mathcal{L}\}$  for any  $\mathcal{L} \subset \Sigma^*$ .

Suppose that a word  $U$  is not  $\widetilde{AB}$ -whole and an almost overlap-free word  $V$  is equivalent to  $U$ . Then  $V$  is not  $\widetilde{AB}$ -whole as well by Proposition 1.3; in particular,  $V$  is non-uniform. Define

$$\begin{aligned} \mathcal{A}_1 = \{ & aabaa, aabaab, baabaa, baabaab, aabaabb, \\ & bbaabaa, aabaaba, abaabaa, aabaabbaabaa \}, \end{aligned}$$

and let  $\mathcal{S}_1 = \mathcal{A}_1 \cup \overline{\mathcal{A}_1}$ . The non-uniform almost overlap-free words can be characterized as follows.

**Proposition 1.6.** *Let  $V$  be a non-uniform  $r_1$ -reduced almost overlap-free word. Then at least one of the following conditions holds:*

- 1)  $V \in \mathcal{S}_1$ ;
- 2) Up to negation,  $V$  has the prefix  $aabaabba$  or the suffix  $abbaabaa$ .

Since all words from  $\mathcal{S}_1$  are almost overlap-free, not uniform, and not  $\widetilde{AB}$ -whole, we immediately get the following corollary of Proposition 1.6.

**Corollary 1.1.** *Any almost overlap-free word is  $\widetilde{AB}$ -whole if and only if it is uniform.*

The  $r_1$ -reduced words that are equivalent to one of the prefixes or suffixes mentioned in Proposition 1.6, 2) will be called *non-uniform tails*. There are four kinds of such tails:

	left	right
A-tail	$(aab)^*(aab)^2ba$	$ab(baa)^2(baa)^*$
B-tail	$(bba)^*(bba)^2ab$	$ba(abb)^2(abb)^*$

The *tail reduction* operation  $r_T$ , defined in [13], reduces a left non-uniform tail of any word to the last 7 symbols, and a right non-uniform tail to the first 7 symbols:

	left	right
reduced A-tail	$abaabba$	$abbaaba$
reduced B-tail	$babbaab$	$baabbab$

Note that a tail of an almost overlap-free word, if any, has exactly 8 symbols. Hence the operation  $r_T$  deletes exactly one letter from such a tail.

In addition to the function  $r_T$ , we use the functions  $r_T^l$  and  $r_T^r$ , which reduce a left (resp., right) non-uniform tail of any given word to the last (resp., first) 7 symbols. If a word  $W$  has both left and right tails, then these tails have at most 6 symbols in common (up to negation, for the word  $W = (aab)^*aabaabbabb(abb)^*$ ). Hence the operation  $r_T^l$  preserves the right tail of  $W$  and  $r_T^r$  preserves the left tail of  $W$ . Thus we have  $r_T(W) = r_T^r(r_T^l(W)) = r_T^l(r_T^r(W))$  for any word  $W$ .

We make one easy observation.

**Observation 1.2.** *Let  $U$  and  $V$  be  $r_1$ -reduced words and  $r_T(U) \sim r_T(V)$ . Then  $U \sim V$ .*

Note that the inverse is not true in general. Surprisingly, under certain conditions, the operation  $r_T$  preserves the congruence  $\sim$ .

**Proposition 1.7.** *Suppose that  $U \sim V$  for  $r_1$ -reduced words  $U$  and  $V$ . Then*

- 1)  $U$  and  $V$  have non-uniform tails of the same kind, if any;
- 2) If  $U, V \notin [aabaabbaabaa]_{r_1} \cup [bbabbaabbabb]_{r_1}$  and at least one of the words  $r_T(U)$  and  $r_T(V)$  is  $\widetilde{AB}$ -whole, then  $r_T(U) \sim r_T(V)$ .

Combining Propositions 1.7, 1.6, 1.3, and the definition of  $r_T$ , we get the following corollary.

**Corollary 1.2.** *Let  $U \sim V$  for an  $r_1$ -reduced word  $U$  and an  $r_1$ -reduced almost overlap-free word  $V$ . If  $V \notin \mathcal{S}_1$ , then  $r_T(U) \sim r_T(V)$  and the words  $r_T(U), r_T(V)$  are  $\widetilde{AB}$ -whole.*

So, it remains to calculate the equivalence classes of all words from  $\mathcal{S}_1$ . The following lemma gives the answer.

**Lemma 1.3.** 1)  $[aabaab]_{r_1} = aabaa;$

2)  $[aabaab]_{r_1} = (aab)^2(aab)^*;$

3)  $[baabaa]_{r_1} = (baa)^*(baa)^2;$

4)  $[baabaab]_{r_1} = (baa)^2(baa)^*b;$

5)  $[aabaabb]_{r_1} = (aab)^2(aab)^*b;$

6)  $[bbaabaa]_{r_1} = b(baa)^*(baa)^2;$

7)  $[aabaaba]_{r_1} = (aab)^2(aab)^*a;$

8)  $[abaabaa]_{r_1} = a(baa)^*(baa)^2;$

9)  $[aabaabbaabaa]_{r_1} = (aab)^2(aab)^*(b(aab)^*aab)^*(baa)^*(baa)^2.$

The classes of the words from  $\overline{\mathcal{A}}_1$  are negations of the classes 1)–9).

Note that for any word  $V \in \mathcal{S}_1$  the equivalence class  $[V]_{r_1}$  is a regular language. Thus the word problem is decidable for any pair  $(U, V)$ : one can build a finite automaton recognizing  $[V]_{r_1}$  and try to accept  $U$  by this automaton.

**1.5 From uniform words to  $\varphi$ -images: functions  $\xi$  and  $\eta$ .** Combining the functions  $r_T$  and  $r$ , we can transform a given word  $U$  to a uniform word. Here we introduce the functions  $\xi$  and  $\eta$ , which transform any uniform word to a  $\varphi$ -image. By Proposition 1.2, any uniform word  $U$  is a factor of a  $\varphi$ -image. That is,

$$U = cQ_1 \dots Q_k d, \text{ where } Q_1, \dots, Q_k \in \{ab, ba\}, c, d \in \{a, b, \lambda\}.$$

This representation is unique if  $U$  is not letter-alternating. For letter-alternating words we additionally require  $c = \lambda$  to get a unique representation as well. Now put

$$\eta(U) = Q_1 \dots Q_k, \quad \xi(U) = \overline{c}U\overline{d}.$$

Obviously,  $\eta(U)$  is a  $\varphi$ -image of maximum length contained in  $U$ , while  $\xi(U)$  is a  $\varphi$ -image of minimum length containing  $U$ . We denote  $h_\eta(U) = c$ ,  $t_\eta(U) = d$ ,  $h_\xi(U) = \overline{c}$ , and  $t_\xi(U) = \overline{d}$ . First, we establish some basic properties of  $\xi$  and  $\eta$ .

**Lemma 1.4.** 1) Let  $eUf \in \varphi(\Sigma^*)$ , where  $e, f \in \{a, b, \lambda\}$  and  $U$  is not letter-alternating. Then  $\xi(U) = eUf$ ,  $e = h_\xi(U)$ , and  $f = t_\xi(U)$ .

2) Let  $U = cU'd$ , where  $c, d \in \{a, b, \lambda\}$  and  $U' \in \varphi(\Sigma^*)$ . Then  $U$  is uniform. Moreover, if  $U$  is not letter-alternating, then  $\eta(U) = U'$ ,  $h_\eta(U) = c$ , and  $t_\eta(U) = d$ ;

3) Let  $U = cU'd$ , where the word  $U'$  is uniform and  $c, d \in \{a, b, \lambda\}$  such that  $c \neq U[1]$ ,  $d \neq U[|U|]$ . Then the word  $U$  is uniform as well.

**Proof.** To prove 2), note that  $U = cU'd$  and  $U' \in \varphi(\Sigma^*)$  imply  $U \leq \overline{c}cU'd\overline{d} \in \varphi(\Sigma^*)$ . Hence the word  $U$  is uniform. If  $U$  is not letter-alternating, then the representation  $cU'd$  is unique, so we are done. Statement 3) of the lemma is evident. Indeed, from  $c \neq U'[1]$  and  $d \neq U'[|U'|]$  it follows that all factors  $aa$  or  $bb$  occur in  $U$  inside the factor  $U'$  only. Adding the symbol  $c$  to the beginning of  $U'$ , we change (if  $c \neq \lambda$ ) or save (if  $c = \lambda$ ) parity of all positions in the word  $U'$ . Therefore, since  $U'$  is uniform, the word  $U$  is uniform as well. Finally, one can see that if  $eUf = \varphi(X)$  for some  $X \in \Sigma^*$ , then  $U = \overline{e}\varphi(X')\overline{f}$ , where  $X' \leq X$ . In view of the uniqueness of such representation, we conclude that  $\eta(U) = \varphi(X')$ ,  $h_\eta(U) = \overline{e}$ , and  $t_\eta(U) = \overline{f}$ . From the definitions of  $h_\xi$  and  $t_\xi$  it immediately follows that  $h_\xi(U) = e$  and  $t_\xi(U) = f$ . The proof is complete.  $\square$

The following observation describes the equivalence classes of all letter-alternating words.

**Observation 1.3.** *The equivalence classes containing letter-alternating words are:  $[a] = a$ ,  $[ab] = ab$ ,  $[aba] = aba$ ,  $[abab] = (ab)^2(ab)^*$ ,  $[ababa] = (ab)^2(ab)^*a$ , and their negations.*

Now we prove that the functions  $h_\xi$ ,  $t_\xi$ ,  $h_\eta$ , and  $t_\eta$  are invariant under the congruence  $\sim$ .

**Proposition 1.8.** *Let  $U, V$  be uniform words and  $U \sim V$ . Then*

- 1)  $h_\xi(U) = h_\xi(V)$ ,  $t_\xi(U) = t_\xi(V)$ ;
- 2)  $h_\eta(U) = h_\eta(V)$ ,  $t_\eta(U) = t_\eta(V)$ .

**Proof.** If  $U$  or  $V$  is letter-alternating, then the proposition readily follows from Observation 1.3. Assume that they are not letter-alternating.

Let us prove the first statement. By definition of congruence,  $U \sim V$  implies  $h_\xi(U)Ut_\xi(U) \sim h_\xi(U)Vt_\xi(U)$ . From  $U \sim V$  and the definition of  $\xi$  it follows that  $h_\xi(U) \neq U[1] = V[1]$  and  $t_\xi(U) \neq U[|U|] = V[|V|]$ . Hence, by Lemma 1.4, 3), the word  $h_\xi(U)Vt_\xi(U)$  is uniform. So we have  $\xi(U) \sim h_\xi(U)Vt_\xi(U)$ , where  $\xi(U) \in \varphi(\Sigma^*)$  and the word  $h_\xi(U)Vt_\xi(U)$  is uniform. According to Proposition 1.1, we get  $h_\xi(U)Vt_\xi(U) \in \varphi(\Sigma^*)$  as well. In view of Lemma 1.4, 1), we have  $h_\xi(V) = h_\xi(U)$  and  $t_\xi(V) = t_\xi(U)$ , as desired.

Statement 2 is trivially follows from statement 1 and the definitions of  $h_\eta$ ,  $t_\eta$ ,  $h_\xi$ , and  $t_\xi$ .  $\square$

According to Proposition 1.8, the function  $\xi$  preserves the congruence  $\sim$  whereas the function  $\eta$  does not preserve  $\sim$  in general case. However, as we will show in Sect. 5, under certain conditions  $\eta$  preserves  $\sim$  as well. The function  $\xi$  is used in the proof of Theorem 1, while the function  $\eta$  plays a crucial role in the construction of Algorithm EqAOF.

Finally, we note that the functions  $\varphi^{-1}(\xi(V))$  and  $\varphi^{-1}(\eta(V))$  preserve the property of a word to be almost overlap-free. For  $\xi$  this was proved in [14], for  $\eta$  even a stronger assertion holds.

**Observation 1.4.** *If  $V$  is almost overlap-free, then the word  $\varphi^{-1}(\eta(V))$  is overlap-free.*

**Proof.** An almost overlap-free word that is not overlap-free has the form  $cYcYc$  for some  $c \in \Sigma$  and  $Y \in \Sigma^*$ , hence, it has odd length. Since the length of  $\eta(V)$  is even,  $\eta(V)$  is a proper factor of  $V$  and, hence, it is overlap-free. The function  $\varphi^{-1}$  preserves overlap-freeness, so the observation follows.  $\square$

## 2 Proof of Theorem 1

We prove Theorem 1 in the form  $(*)$  and use the minimal counterexample method. Suppose that a pair  $(U, V)$  provides a counterexample, i. e.,  $U$  and  $V$  are two nonequal  $r_1$ -reduced almost overlap-free words such that  $U \sim V$ , and  $l = \min\{|U|, |V|\}$  takes the minimum value among all such pairs. We aim to get a contradiction by obtaining a “shorter” counterexample.

Clearly,  $l > 2$ . By Lemma 1.3, any equivalence class  $[W]_{r_1}$  for  $W \in \mathcal{S}_1$  contains exactly one almost overlap-free word (namely, the word  $W$  itself).

Therefore  $U, V \notin \mathcal{S}_1$ . Similarly, by Observation 1.3, the words  $U$  and  $V$  are not letter-alternating. It follows from Corollaries 1.1 and 1.2 that  $r_T(U) \sim r_T(V)$  and  $r_T(U), r_T(V)$  are uniform almost overlap-free words. Hence, by Proposition 1.8, the words  $U_1 = \varphi^{-1}(\xi(r_T(U)))$  and  $V_1 = \varphi^{-1}(\xi(r_T(V)))$  are equivalent. Obviously, both words  $U_1$  and  $V_1$  are almost overlap-free and  $U_1 \neq V_1$ .

Since the words  $U$  and  $V$  are not letter-alternating, the words  $\xi(r_T(U))$  and  $\xi(r_T(V))$  are not letter-alternating as well. In particular, both words  $\xi(r_T(U))$  and  $\xi(r_T(V))$  are not equal to  $ababab$  or  $bababa$ . Therefore  $U_1, V_1 \notin \{aaa, bbb\}$ .

Finally, we have  $l_1 = \min\{|U_1|, |V_1|\} \leq (l+2)/2 < l$  whenever  $l > 2$ . So the pair  $(U_1, V_1)$  provides a shorter counterexample. This completes the proof of Theorem 1.

### 3 Procedure Ancestor

In this section we begin the construction of Algorithm AqEOF, which returns the almost overlap-free word that is equivalent to a given word or reports that no such almost overlap-free word exists. Algorithm AqEOF includes two main procedures. Let us define the first of them.

#### Procedure Ancestor.

*Input.* A word  $U \in \Sigma^+$ .

*Output.* A word  $\text{Anc}(U) \in \Sigma^+$ , integer  $k$ , length  $k$  arrays  $L, R, h, t$  of letters.

**Step 0.** Let  $k := 0$ .

**Step 1.** Let  $U = r_1(U)$ ;  $k = k + 1$ ,  $L[k] = R[k] = h[k] = t[k] := \lambda$ .

**Step 2.** If  $|U| \leq 2$  or  $U \in [aabaabbaabaa] \cup [bbabbaabbabb]$ , then  $\text{Anc} := U$ ; stop.

**Step 3.** If  $U$  has a non-uniform left (right) tail, set  $L[k] := U[1]$  (resp.,  $R[k] = U[|U|]$ ); let  $U' := r_T(U)$ .

**Step 4.** If  $U'$  is not  $\widehat{AB}$ -whole or  $U'$  has a non-reducible tail, then  $\text{Anc} := U$ ; stop.

**Step 5.** Let  $U' := r(U')$ .

**Step 6.** Let  $h[k] := h_\eta(U')$ ;  $t[k] := t_\eta(U')$ ;  $U := \varphi^{-1}(\eta(U'))$ ; goto step 1.

**End.**

Starting with  $U_1 = r_1(U)$ , procedure Ancestor constructs the sequence of words  $U_1, U_2, \dots$  by the rule:

$$U_{k+1} = r_1(\varphi^{-1}(\eta(r(r_T(U_{k-1}))))))$$

until one of the stop conditions is fulfilled. We call the sequence  $\{U_k\}$  the *primary  $U$ -series*. Since  $|U_k| < |U_{k-1}|/2$  for any  $k \geq 2$ , the primary  $U$ -series is finite, its *length* (that is, the number of words in it) is denoted by  $\ell(U)$ . We say that the output word  $\text{Anc}(U) = U_{\ell(U)}$  is the *ancestor of  $U$* , and the arrays  $L = L_U, R = R_U, h = h_U$ , and  $t = t_U$  returned by Procedure Ancestor are *associated with  $U$* . We omit the index  $U$  if it is clear from context.

The next two lemmas establish the basic properties of primary series. First, we examine primary series of almost overlap-free words.

**Lemma 3.1.** *Let  $V$  be an almost overlap-free word and  $\{V_k\}_{k=1}^{\ell(V)}$  be its primary  $V$ -series. Then*

- 1)  $\text{Anc}(V) \in \mathcal{S}_1 \cup \{a, b, ab, ba, aa, bb\}$ ;
- 2)  $V_k$  is overlap-free for each  $k = 2, \dots, \ell(V)$ ;
- 3)  $V_k = L[k]r(r_T(V_k))R[k]$  for each  $k = 1, \dots, \ell(V)-1$ .

**Proof.** Instead of 1)–3) we prove the following statement:

*if for some  $k \geq 1$  the word  $V_k$  is almost overlap-free,  $|V_k| > 2$ , and  $V_k \notin \mathcal{S}_1$ , then  $V_k = L[k]r(r_T(V_k))R[k]$ ,  $k < \ell(V)$ , and the word  $V_{k+1}$  is overlap-free.*

Consider the  $k$ th iteration of procedure Ancestor, i. e., the processing of the word  $V_k$ . By the conditions above, procedure Ancestor cannot stop on Step 2. By Proposition 1.6,  $r_T(V_k)$  is a uniform almost overlap-free word. Hence procedure Ancestor cannot stop on Step 4 as well. Thus procedure Ancestor does not stop on the  $k$ th iteration, so we have  $k < \ell(V)$ , and the word  $V_{k+1} = r_1(\varphi^{-1}(\eta(r_T(V_k))))$  is overlap-free in view of Observation 1.4. From the uniformity of  $r_T(V_k)$  it follows that  $r(r_T(V_k)) = r_T(V_k)$ . Since the function  $r_T$  deletes at most one letter from the beginning and at most one letter from the end of any almost overlap-free word (see the remark after the definition of  $r_T$ ), we get  $V_k = L[k]r(r_T(V_k))R[k]$ , as desired.  $\square$

Now we apply procedure Ancestor to a pair  $(U, V)$  of equivalent words.

**Lemma 3.2.** *Suppose that  $U \sim V$ ,  $m = \min\{\ell(U), \ell(V)\}$ ,  $\{U_k\}_{k=1}^{\ell(U)}$ ,  $\{V_k\}_{k=1}^{\ell(V)}$  are the primary  $U$ - and  $V$ -series respectively, and the following condition:*

$$(**) \quad r(r_T(U_k)) \sim r(r_T(V_k)) \Rightarrow \eta(r(r_T(U_k))) \sim \eta(r(r_T(V_k))) \text{ for all } k < m$$

*holds. Then*

- 1)  $\ell(U) = \ell(V)$ ;
- 2)  $U_k \sim V_k$  for each  $k = 1, \dots, m$ ; in particular,  $\text{Anc}(U) \sim \text{Anc}(V)$ ;
- 3)  $L_U = L_V$ ,  $R_U = R_V$ ,  $h_U = h_V$ , and  $t_U = t_V$ .

**Proof.** Instead of 1)–3) we prove the following statement:

*if  $U_k \sim V_k$  for some  $k \leq m$ , then  $L_U[k] = L_V[k]$ ,  $R_U[k] = R_V[k]$ ,  $h_U[k] = h_V[k]$ ,  $t_U[k] = t_V[k]$ , and either  $k = m = \ell(U) = \ell(V)$  or  $k < m$  and  $U_{k+1} \sim V_{k+1}$ .*

First, suppose that  $k = m = \ell(U)$ . If procedure Ancestor stops to process the word  $U_k$  on Step 2, then the processing of the word  $V_k$  should stop on Step 2 as well, because  $U_k \sim V_k$  and both words are  $r_1$ -reduced. In this case, we have  $\ell(U) = \ell(V)$ , and the  $k$ th elements of all arrays associated with  $U$  and  $V$  are equal to  $\lambda$ .

Now suppose that procedure Ancestor stops to process the word  $U_k$  on Step 4. Assume additionally that the word  $\underline{V_k}$  is  $\widetilde{AB}$ -whole and has no non-reducible tails. Then the word  $r_T(V_k)$  is  $\widetilde{AB}$ -whole as well and we have  $r_T(U_k) \sim r_T(V_k)$  by Proposition 1.7, 2). At the same time, from Lemma 1.1 and Proposition 1.3 it follows that the word  $r_T(U_k)$  is  $\widetilde{AB}$ -whole and has no non-reducible tails as well as  $r_T(V_k)$ . So procedure Ancestor cannot stop to process the word  $U_k$  on Step 4, a contradiction. Hence either the word  $V_k$  is not  $\widetilde{AB}$ -whole or it has a non-reducible tail. In both cases procedure Ancestor

stops to process the word  $V_k$  on Step 4, and we get  $k = \ell(U) = \ell(V)$ . Also,  $L_U[k] = L_V[k]$ ,  $R_U[k] = R_V[k]$  by Proposition 1.7, 1) and  $h_U[k] = t_U[k] = h_V[k] = t_V[k] = \lambda$ .

The case  $k = m = \ell(V)$  is symmetrical to above one. Finally, suppose that  $k < m$ . We have  $r(r_T(U_k)) \sim r(r_T(V_k))$  by Propositions 1.4 and 1.7. From condition (\*\*) it now follows that  $\eta(r(r_T(U_k))) \sim \eta(r(r_T(V_k)))$  whence

$$U_{k+1} = r_1(\varphi^{-1}(\eta(r(r_T(U_k))))) \sim r_1(\varphi^{-1}(\eta(r(r_T(V_k))))) = V_{k+1}.$$

In additional, we have  $L_U[k] = L_V[k]$ ,  $R_U[k] = R_V[k]$  by Proposition 1.7, 1) and  $h_U[k] = h_V[k]$ ,  $t_U[k] = t_V[k]$  by Proposition 1.8. This completes the proof of the statement.  $\square$

As we said above, under certain conditions, the function  $\eta$  preserves the congruence  $\sim$ . In the sequel, we call a pair  $(P, Q)$  of equivalent uniform words *good* if  $\eta(P) \sim \eta(Q)$ , and *bad* otherwise. So, condition (\*\*) provides that all pairs  $(r(r_T(U_k)), r(r_T(V_k)))$  are good. We set the study of bad pairs aside for Sect. 5.

## 4 Normal series

The notion in the headline plays a crucial role in the second main procedure of Algorithm EqAOF (this procedure will be presented in Sect. 6). This notion is defined as follows.

Take a word  $U$  and a cube-free word  $W \in [\text{Anc}(U)]$ . The *normal  $U_W$ -series* or the  *$W$ -normal series of  $U$*  is the sequence of words  $\tilde{U}_{\ell(U)}, \tilde{U}_{\ell(U)-1}, \dots, \tilde{U}_1$  defined by

$$\tilde{U}_{\ell(U)} = W, \quad \tilde{U}_k = L_U[k]h_U[k]\varphi(\tilde{U}_{k+1})t_U[k]R_U[k], \quad k = \ell(U)-1, \dots, 1.$$

The word  $\tilde{U}_1$  is called the  *$W$ -normal* (or simply a *normal*) *form of the word  $U$*  and denoted by  $N_W(U)$ . Clearly, any word  $U$  has at least one normal series, since the class  $[\text{Anc}(U)]$  contains cube-free words. However, the class  $[\text{Anc}(U)]$  can contain several cube-free words, therefore  $U$  can have several normal series. If  $W$  is almost overlap-free, then the normal  $U_W$ -series is the *main normal  $U$ -series*, the  $W$ -normal form of  $U$  is called the *main normal form of  $U$*  and denoted by  $N(U)$ . By Theorem 1, each word  $U$  has at most one main normal series. A normal series or main normal series is called *direct* if  $W = \text{Anc}(U)$ . In this case, the normal form  $N_W(U)$  is called *direct* as well and denoted by  $N^D(U)$ .

First, we prove the main property of normal forms.

**Lemma 4.1.** *Suppose that  $W \sim \text{Anc}(U)$  for a word  $U$  and a cube-free word  $W$ . Let  $\{U_k\}_{k=1}^{\ell(U)}$  and  $\{\tilde{U}_k\}_{k=1}^{\ell(U)}$  be the primary  $U$ -series and the normal  $U_W$ -series respectively. Then  $\tilde{U}_k \sim U_k$  for each  $k = 1, \dots, \ell(U)$ ; in particular,  $N_W(U) \sim U$ .*

**Proof.** For  $k = \ell(U)$ , we have  $\tilde{U}_{\ell(U)} = W$  and  $U_{\ell(U)} = \text{Anc}(U)$  from definitions. Now suppose that  $\tilde{U}_{k+1} \sim U_{k+1}$  for some  $k < \ell(U)$ . If we show that  $\tilde{U}_k \sim U_k$ , the required statement will follow by induction. We have

$$\tilde{U}_{k+1} \sim U_{k+1} = r_1(\varphi^{-1}(\eta(r(r_T(U_k)))))$$

whence we get  $\tilde{U}_{k+1} \sim \varphi^{-1}(\eta(r(r_T(U_k))))$  and  $h[k]\varphi(\tilde{U}_{k+1})t[k] \sim r(r_T(U_k))$ . Since procedure Ancestor does not stop on  $k$ th iteration, the word  $r_T(U_k)$  is  $\overline{AB}$ -whole and has no non-reducible tails. From Proposition 1.4 it now follows that  $r(r_T(U_k)) \sim r_T(U_k)$ . Hence,  $h[k]\varphi(\tilde{U}_{k+1})t[k] \sim r_T(U_k)$ . Note that if  $U_k$  has non-uniform tails, then the word  $L[k]r_T(U_k)R[k]$  has non-uniform tails of the same kind as the word  $U_k$ . Since all non-uniform tails of the same kind are equivalent, we conclude that  $L[k]r_T(U_k)R[k] \sim U_k$ . Thus, we have

$$\tilde{U}_k = L[k]h[k]\varphi(\tilde{U}_{k+1})t[k]R[k] \sim L[k]r_T(U_k)R[k] \sim U_k,$$

as desired.  $\square$

So, normal series are “inverted” to primary series in a sense. Namely, by Lemma 4.1, if a cube-free word  $W$  is equivalent to  $\text{Anc}(U)$ , then the  $W$ -normal series of  $U$  allows one to restore the primary series  $\{U_k\}_{k=1}^{\ell(U)}$  up to equivalent words. Actually, the  $U_W$ -normal series consists of the words  $\tilde{U}_k$  with more simple structure than  $U_k$  in the general case. The following lemma describes the structure of the words  $\tilde{U}_k$ .

**Lemma 4.2.** *Suppose that  $\text{Anc}(U) \sim W$  for a word  $U$  and a cube-free word  $W$ . Let  $\{\tilde{U}_k\}_{k=0}^{\ell(U)}$  be the  $W$ -normal series of  $U$ . Then*

- 1)  $\tilde{U}_k$  is an  $r_1$ -reduced word for each  $k = 1, \dots, \ell(U)$ ;
- 2)  $r_T(\tilde{U}_k)$  is uniform for each  $k = 1, \dots, \ell(U)-1$ ;
- 3)  $r_T(\tilde{U}_k) = h[k]\varphi(\tilde{U}_{k+1})t[k]$  for each  $k = 1, \dots, \ell(U)-1$ ;
- 4)  $h(r_T(\tilde{U}_k)) = h[k]$ ,  $t(r_T(\tilde{U}_k)) = t[k]$ , and  $\eta(r_T(\tilde{U}_k)) = \varphi(\tilde{U}_{k+1})$  for each  $k = 1, \dots, \ell(U)-1$ .

**Proof.** For  $k = \ell(U)$ , there is nothing to prove. Suppose  $k < \ell(U)$ . Let us denote the word  $h[k]\varphi(\tilde{U}_{k+1})t[k]$  by  $\tilde{U}'_k$ , and let  $\{U_k\}_{k=1}^{\ell(U)}$  be the primary  $U$ -series. The word  $\tilde{U}'_k$  is uniform by Lemma 1.4, 2). Since  $\tilde{U}_{k+1} \sim U_{k+1} \sim \varphi^{-1}(\eta(r(r_T(U_k))))$  by Lemma 4.1, we get  $\tilde{U}'_k \sim r(r_T(U_k))$ . From Proposition 1.8 it now follows that  $h(\tilde{U}'_k) = h[k]$ ,  $t(\tilde{U}'_k) = t[k]$ , and  $\eta(\tilde{U}'_k) = \varphi(\tilde{U}_{k+1})$ .

It remains to prove that  $\tilde{U}_k$  is an  $r_1$ -reduced word and  $r_T(\tilde{U}_k) = \tilde{U}'_k$ . We have  $\tilde{U}_k = L[k]\tilde{U}'_kR[k]$ . First, we apply the function  $r_T^l$  to the word  $L[k]\tilde{U}'_k$ . Clearly, if  $L[k] = \lambda$ , then  $r_T^l(L[k]\tilde{U}'_k) = \tilde{U}'_k$ . Now suppose that  $L[k] \neq \lambda$ . Without loss of generality we assume that  $L[k] = a$ . This means that  $U_k$  has the non-uniform left tail  $(aab)^k(aab)^2ba$  for some  $k \geq 0$ . Hence the word  $r(r_T(U_k))$  has the prefix  $abaabba$ . Therefore the word  $\tilde{U}'_k$ , which is equivalent to  $r(r_T(U_k))$ , begins with  $abaabba$  as well. Since the word  $L[k]\tilde{U}'_k$  does not begin with  $aaa$ , it is  $r_1$ -reduced. So we obtain  $r_T^l(L[k]\tilde{U}'_k) = \tilde{U}'_k$ .

In the symmetric way one can prove that  $\tilde{U}'_kR[k]$  is an  $r_1$ -reduced word and  $r_T^r(\tilde{U}'_kR[k]) = \tilde{U}'_k$ . So, the word  $\tilde{U}_k = L[k]\tilde{U}'_kR[k]$  is  $r_1$ -reduced and we have

$$r_T(\tilde{U}_k) = r_T^r(r_T^l(L[k]\tilde{U}'_kR[k])) = r_T^r(\tilde{U}'_kR[k]) = \tilde{U}'_k.$$

This completes the proof.  $\square$

Next we investigate primary series and normal series for normal forms.

**Lemma 4.3.** *Let  $\text{Anc}(U) \sim W$  for a word  $U$  and a cube-free word  $W$ . Then both the primary  $N_W(U)$ -series and the  $W$ -normal series of  $N_W(U)$  coincide with the  $U_W$ -normal series.*



**Proof.** Let us denote  $P = N_W(U)$ ,  $m = \ell(U)$ ,  $l = \ell(P)$ , and let  $\{U_k\}_{k=1}^m$ ,  $\{\tilde{U}_k\}_{k=1}^m$ , and  $\{P_k\}_{k=1}^l$  be the primary series of  $U$ , the  $U_W$ -normal series, and the primary series of  $P$  respectively. First, we prove that the primary  $P$ -series coincides with the normal  $U_W$ -series.

By Lemma 4.2, the word  $P$  is  $r_1$ -reduced. Hence,  $P_1 = P = N_W(U) = \tilde{U}_1$ . Now suppose that  $P_k = \tilde{U}_k$  for some  $k < m$ . In view of  $\tilde{U}_k \sim U_k$  and  $k < \ell(U)$ , we conclude that  $|P_k| > 2$  and  $P_k \notin [aabaabbaabaa] \cup [bbabbaabbb]$ . In addition, the word  $r_T(P_k)$  is uniform by Lemma 4.2. Hence procedure Ancestor does not stop while processing the word  $P_k$ , that is,  $k < l$ , and we have  $r(r_T(P_k)) = r_T(\tilde{U}_k)$ . Finally, by Lemma 4.2 we get

$$\begin{aligned} P_{k+1} &= r_1(\varphi^{-1}(\eta(r(r_T(P_k))))) = r_1(\varphi^{-1}(\eta(r_T(\tilde{U}_k)))) = \\ &= r_1(\varphi^{-1}(\varphi(\tilde{U}_{k+1}))) = r_1(\tilde{U}_{k+1}) = \tilde{U}_{k+1}, \end{aligned}$$

as desired.

Now consider the word  $P_m = \tilde{U}_m = W \sim \text{Anc}(U)$ . One can easily check that if procedure Ancestor stops processing the word  $\text{Anc}(U)$  on Step 2 or Step 4, then it stops processing the word  $P_m$  on Step 2 or respectively Step 4 as well. Thus,  $l = m$ , and the primary  $P$ -series coincides with the normal  $U_W$ -series.

Clearly, since  $P_k = \tilde{U}_k \sim U_k$ , we have  $L_U[k] = L_P[k]$ ,  $R_U[k] = R_P[k]$ ,  $h_U[k] = h_P[k]$ , and  $t_U[k] = t_P[k]$  for each  $k = 1, \dots, m$ . This implies that the  $P_W$ -normal series coincides with the  $U_W$ -normal series. The proof is complete.  $\square$

In particular, it follows from Lemma 4.3 that  $N_W(N_W(U)) = N_W(U)$  for any word  $U$  and any cube-free word  $W$  such that  $W \sim \text{Anc}(U)$ . So, the repeated use of the primary and normal  $U$ -series has no effect.

Finally, we study the main normal series for almost overlap-free words.

**Lemma 4.4.** *If  $U$  is an almost overlap-free word and  $U \notin \{aaa, bbb\}$ , then there exists the main direct normal  $U$ -series and it coincides with the primary  $U$ -series; in particular,  $N(U) = U$ .*

**Proof.** Let  $\{U_k\}_{k=1}^{\ell(U)}$  and  $\{\tilde{U}_k\}_{k=1}^{\ell(U)}$  be the primary and the direct normal  $U$ -series respectively. We prove the lemma by induction on  $k = \ell(U), \dots, 2, 1$ . For  $k = \ell(U)$ , we have  $\tilde{U}_{\ell(U)} = \text{Anc}(U) = U_{\ell(U)}$  by the definition of direct series. Note that since the word  $\text{Anc}(U)$  is almost overlap-free by Lemma 3.1, the direct normal series  $\{\tilde{U}_k\}_{k=1}^{\ell(U)}$  is the main direct normal series of  $U$ .

Now suppose that  $\tilde{U}_k = U_k$ , where  $1 < k \leq \ell(U)$ , and prove that  $\tilde{U}_{k-1} = U_{k-1}$ . Let  $U'_k = \eta(r(r_T(U_{k-1})))$ . First, we prove that the word  $\varphi^{-1}(U'_k)$  is  $r_1$ -reduced. Indeed, if  $aaa \leq \varphi^{-1}(U'_k)$  (or  $bbb \leq \varphi^{-1}(U'_k)$ ), then  $ababab \leq U'_k$  (resp.,  $bababa \leq U'_k$ ). However, it is impossible, since  $U'_k \leq r(r_T(U_{k-1})) \leq U_{k-1}$  by Lemma 3.1 and  $U_{k-1}$  is almost overlap-free. Hence the word  $\varphi^{-1}(U'_k)$  is  $r_1$ -reduced and  $U_k = r_1(\varphi^{-1}(U'_k)) = \varphi^{-1}(U'_k)$ . Therefore we obtain

$$\begin{aligned} U_{k-1} &= L[k-1]r(r_T(U_{k-1}))R[k-1] \quad (\text{by Lemma 3.1}) \\ &= L[k-1]h[k-1]U'_k t[k-1]R[k-1] = L[k-1]h[k-1]\varphi(U_k) t[k-1]R[k-1] \\ &= L[k-1]h[k-1]\varphi(\tilde{U}_k) t[k-1]R[k-1] = \tilde{U}_{k-1}, \end{aligned}$$

as desired.  $\square$

Lemmas 3.1, 3.2, and 4.4 together provide the following assertion:

*if a word  $U$  is equivalent to an almost overlap-free word  $V \notin \{aaa, bbb\}$  and condition  $(**)$  holds, then  $V = N(U)$ .*

Indeed, from Lemma 3.2 it follows that  $\text{Anc}(U) \sim \text{Anc}(V)$ ,  $L_U = L_V$ ,  $R_U = R_V$ ,  $h_U = h_V$ , and  $t_U = t_V$ . By Lemma 3.1, the word  $\text{Anc}(V)$  is almost overlap-free, in particular, it is cube-free. Thus the  $\text{Anc}(V)$ -normal series of the words  $U$  and  $V$  coincide. Since the word  $\text{Anc}(V)$  is almost overlap-free, these normal series are the main normal series of the words  $U$  and  $V$ . Hence,  $N(U) = N(V)$ . By Lemma 4.4, we get  $V = N(V) = N(U)$ , as desired.

We conclude this section explaining why do we use the function  $\eta$ , not  $\xi$ , for the construction of Algorithm AqEOF. Replacing  $\eta$  by  $\xi$  in procedure Ancestor, we simplify the formulation of Lemma 3.2, since condition  $(**)$  can be replaced by the following condition, which is obviously true:

$$r(r_T(U_k)) \sim r(r_T(V_k)) \Rightarrow \xi(r(r_T(U_k))) \sim \xi(r(r_T(V_k))) \text{ for all } k < m,$$

where  $m = \min\{\ell(U), \ell(V)\}$ . Clearly, Lemmas 3.1, 3.2, and 4.4 hold true after replacing  $\eta$  by  $\xi$ . So, if we retain the same notion of normal series, we will get the following assertion:

*if a word  $U$  is equivalent to an almost overlap-free word  $V \notin \{aaa, bbb\}$ , then  $V = N(U)$ .*

Unfortunately, Lemma 4.1 fails if we use  $\xi$  instead of  $\eta$ . Actually, in this case the word  $r_T(\tilde{U}_k)$  will be obtained from  $\varphi(\tilde{U}_{k+1})$  by deleting the symbols  $h[k] = h_\xi(r(r_T(U_k)))$  and  $t[k] = t_\xi(r(r_T(U_k)))$  added to the word  $r(r_T(U_k))$  by procedure Ancestor. However, deleting does not preserve the congruence  $\sim$  in general case. For example, consider the word  $U = bababb$ . Using  $\xi$  instead of  $\eta$ , we get  $N(bababb) = babb$ . The word  $N(U)$  is almost overlap-free, but  $babb \not\sim bababb$ . In fact, the equivalence class  $[bababb]$  contains no almost overlap-free words. So, we simplify the necessary condition for a given word  $U$  to be equivalent to an almost overlap-free word, but we lose the sufficient condition provided by Lemma 4.1. On the other hand, the use of  $\eta$  makes Lemma 4.1 true. This is the main reason why the function  $\eta$ , not  $\xi$ , is used in the construction of Algorithm EqAOF. Surprisingly, as we will prove in the next section, under certain restrictions, the function  $\eta$  preserves the congruence  $\sim$ . Moreover, even if  $U \sim V$  and  $\eta(U) \not\sim \eta(V)$  for some pair  $(U, V)$  (that is, the pair  $(U, V)$  is bad), then an analogue of Lemma 3.2 holds (see Lemma 5.4 below).

## 5 Bad pairs of equivalent uniform words

This section is devoted to the study of bad pairs, for which  $U \sim V$  and  $\eta(U) \not\sim \eta(V)$ . In the sequel, we write  $h(U)$  and  $t(U)$  instead of  $h_\eta(U)$  and  $t_\eta(U)$ .

First, we study bad pairs of neighbours. More precisely, we consider pairs  $(U, V)$  such that  $(U, V) \in \pi$  and  $(\eta(U), \eta(V)) \notin \pi$ .

**Lemma 5.1.** *Suppose  $U$  and  $V$  are uniform,  $|U| < |V|$ ,  $(U, V) \in \pi$ , and  $(\eta(U), \eta(V)) \notin \pi$ . Then  $\{h(U), t(U)\} = \{a, b\}$  and there exists an even-length word  $Y$  such that  $|Y| > 2$ ,  $U = YY$ ,  $V = YYY$ ,  $h(Y) = h(U) = h(V)$ , and  $t(Y) = t(U) = t(V)$ .*

**Proof.** Since  $|U| < |V|$ , we can write  $U = XYYZ$  and  $V = XYYYZ$ , where  $X, Z \in \Sigma^*$  and  $Y \in \Sigma^+$ . We show that the word  $Y$  is the required one.

First, we prove that the length of  $Y$  is even. Assume the converse. If the word  $Y$  is letter-alternating, then  $V$  has the factor  $Y[|Y|]YY[1]$  of the form  $a\tilde{A}a$  or  $b\tilde{B}b$ , a contradiction with the uniformity of  $V$ . Now let  $Y = PccQ$  for some  $P, Q \in \Sigma^*$ , and  $c \in \Sigma$ . Then we have  $ccQPcc \leq YY \leq U$  and  $ccQPcc \leq YY \leq V$ . This contradicts the uniformity of  $U$  and  $V$  again, since the length of  $PQ$  is odd. Hence, the length of  $Y$  is even.

By Proposition 1.8, we have  $h(U) = h(V) = c$  and  $t(U) = t(V) = d$  for some  $c, d \in \{a, b, \lambda\}$ . If  $|c| \leq |X|$  and  $|d| \leq |Z|$ , then the words  $\eta(U)$  and  $\eta(V)$  are neighbours by definition. Hence, at least one of these inequalities should fail.

Suppose that  $|d| \leq |Z|$  and  $|c| > |X|$ , that is,  $c \neq \lambda$  and  $X = \lambda$ . Then  $Y = cY'e$  for some  $Y' \in \Sigma^*$  and  $e \in \Sigma$ . So, we get  $\eta(U) = Y'ecY'eZ'$  and  $\eta(V) = Y'ecY'ecY'eZ'$ , where  $Z' \in \Sigma^*$ . By Observation 1.1, the length of  $Z'$  is odd, in particular,  $Z' \neq \lambda$ . Also, we have  $Z'[1] = \bar{e} = c$ , since the factors  $ec$  and  $eZ'[1]$  start in  $\eta(U)$  at odd positions. Thus the words  $\eta(U) = (Y'ec)^2Z''$  and  $\eta(V) = (Y'ec)^3Z''$ , where  $Z'' = Z'[2 \dots |Z'|]$ , are neighbours, a contradiction.

The case  $|c| \leq |X|$  and  $|d| > |Z|$  is symmetric to the above one. Thus,  $X = Z = \lambda$  and  $c, d \neq \lambda$ . Then  $U = YY$ ,  $V = YYY$ ,  $Y[1] = c$ , and  $Y[|Y|] = d$ . Let  $Y = cY'd$ , where  $Y' \in \Sigma^*$ . Then  $\eta(U) = Y'dcY'$  and  $\eta(V) = Y'dcY'dcY'$ . By Observation 1.1, we have  $Y' \in \varphi(\Sigma^*)$  and  $c = \bar{d}$ . If the word  $Y$  is letter-alternating, then the words  $U$  and  $V$  appear to be  $\varphi$ -images, since the length of  $Y$  is even. However, this contradicts the assumption  $h(U), t(U) \neq \lambda$ . Hence,  $Y$  is not letter-alternating. From Lemma 1.4, 2) it now follows that  $h(Y) = c$  and  $t(Y) = d$ .

Note that  $Y \neq cd = c\bar{c}$ , since the word  $Y$  is not letter-alternating. Therefore,  $|Y| > 2$ . This completes the proof.  $\square$

As a consequence we get the next proposition.

**Proposition 5.1.** *If a pair  $(U, V)$  is bad, then*

$$\{h(U), t(U)\} = \{h(V), t(V)\} = \{a, b\}.$$

**Proof.** Let  $\{W_k\}_{k=0}^n$  be an  $r$ -linking  $(U, V)$ -series. Clearly, if all the pairs  $(W_{k-1}, W_k)$  are good, then the pair  $(U, V)$  is good as well. Hence, there exists an integer  $k' \geq 1$  such that the pair  $(W_{k'-1}, W_{k'})$  is bad. By Lemma 5.1, we have  $\{h(W_{k'}), t(W_{k'})\} = \{a, b\}$ . The required statement now follows from Proposition 1.8.  $\square$

Now consider a bad pair  $(YY, YYY)$ . By Lemma 5.1, we have

$$\varphi^{-1}(\eta(YY)) = \varphi^{-1}(\eta(Y)t(Y)h(Y)\eta(Y)) = \varphi^{-1}(\eta(Y))t(Y)\varphi^{-1}(\eta(Y))$$

and

$$\begin{aligned}\varphi^{-1}(\eta(YYY)) &= \varphi^{-1}(\eta(Y)t(Y)h(Y)\eta(Y)t(Y)h(Y)\eta(Y)) \\ &= \varphi^{-1}(\eta(Y))t(Y)\varphi^{-1}(\eta(Y))t(Y)\varphi^{-1}(\eta(Y)).\end{aligned}$$

So, we get a pair of the form  $(XcX, XcXcX)$ , where  $X \in \Sigma^+$ ,  $c \in \Sigma$ , and  $XcX \not\sim XcXcX$ . For the sequel, we need to establish some useful properties of such pairs.

**Lemma 5.2.** *Let  $XcX \not\sim XcXcX$  for some  $X \in \Sigma^+$ ,  $c \in \Sigma$ . Then*

$$1) \quad r_1(XcX) = r_1(X)cr_1(X), \quad r_1(XcXcX) = r_1(X)cr_1(X)cr_1(X).$$

*Suppose additionally that the words  $XcX$  and  $XcXcX$  are  $r_1$ -reduced. Then*

$$2) \quad \text{If } X \neq \overline{cc}, \text{ then } XcX, XcXcX \notin [W]_{r_1} \text{ for any } W \in \mathcal{S}_1.$$

*3) If one of the words  $XcX$  and  $XcXcX$  is letter-alternating, then all the words  $XcX$ ,  $XcXcX$ , and  $X$  are letter-alternating and have odd length.*

*4) If one of the words  $XcX$  and  $XcXcX$  has a non-uniform tail, then the other has the same non-uniform tail.*

*5) If one of the words  $XcX$  and  $XcXcX$  has a non-reducible tail, then the other has the same non-reducible tail.*

*6) If the word  $XcXcX$  is  $\tilde{A}$ -whole ( $\tilde{B}$ -whole), then both  $X$  and  $XcX$  are  $\tilde{A}$ -whole (resp.,  $\tilde{B}$ -whole). If  $X \notin \{ab, ba\}$  and the word  $XcX$  is  $\tilde{A}$ -whole ( $\tilde{B}$ -whole), then both  $X$  and  $XcXcX$  are  $\tilde{A}$ -whole (resp.,  $\tilde{B}$ -whole).*

*7) If the words  $XcX$  and  $XcXcX$  are  $\widetilde{AB}$ -whole and have no non-reducible tails, then  $r(XcX) = r(X)cr(X)$  and  $r(XcXcX) = r(X)cr(X)cr(X)$ .*

*8) If the words  $XcX$  and  $XcXcX$  are uniform, then the length of  $X$  is odd,  $h(XcX) = h(XcXcX) = h(X)$ , and  $t(XcX) = t(XcXcX) = t(X)$ .*

**Proof.** Prove the first statement of the lemma. Clearly, if

$$aaa \in \{X[|X|-1]X[|X|]a, X[|X|]aX[1], aX[1]X[2]\},$$

then  $XaXaX \sim XXX \sim XX \sim XaX$ , in contradiction with the lemma's condition. Hence, the factors  $aaa$  and  $bbb$  can occur in the words  $XaX$  and  $XaXaX$  inside the factor  $X$  only. This proves the first statement.

Now suppose that both words  $XaX$  and  $XaXaX$  are already  $r_1$ -reduced. Prove the second statement. Assume the converse. Then  $V \in [W]_{r_1}$  for some words  $V \in \{XaX, XaXaX\}$  and  $W \in \mathcal{S}_1$ . Note that the word  $Va$  is either a square or a cube. A straightforward check shows that no words  $V \notin [bbabbaabbabb]_{r_1}$  with this property satisfy the regular expressions listed in Lemma 1.3. Hence,  $W = bbabbaabbabb$ . By Lemma 1.3, the word  $V$  has the following form:

$$(bba)^2(bba)^*(a(bba)^+)^*(abb)^*(abb)^2.$$

In particular, we get  $X[1] = X[|X|] = b$ . Therefore the factor  $aa$  occurs in  $V$  inside the factor  $X$  only. It is easy to check that the word  $X$  has the form  $(bba)^2(bba)^*(a(bba)^+)^*(abb)^+$  as a prefix of  $V$  and has the form  $(bba)^+(a(bba)^+)^*(abb)^*(abb)^2$  as a suffix of  $V$ . Combining these forms, we conclude that  $X$  has the form above. Obviously, the words  $XaX$  and  $XaXaX$  has the same form as well. Thus we get  $X \sim XaX \sim XaXaX$ , in contradiction with the lemma's condition. So,  $V$  is not equivalent to a word from  $\mathcal{S}_1$ .

If the word  $XaXaX$  ( $XaX$ ) is letter-alternating, then the words  $XaX$  and  $X$  (resp.,  $X$ ) are letter-alternating as well. On the other hand, since the factors  $aa$  and  $bb$  occur in  $XaXaX$  inside the factor  $XaX$  only, the word  $XaXaX$  is letter-alternating whenever  $XaX$  is. As it will be shown in the proof of 8), the length of  $X$  is odd whenever the words  $XaX$  and  $XaXaX$  are uniform (in particular, letter-alternating). Hence, the lengths of both words  $XaXaX$  and  $XaX$  are odd as well. The proof of statement 3 is complete.

Prove statements 4–5. Obviously, any tail of the word  $XaX$  (non-reducible or non-uniform) is the tail of  $XaXaX$  as well. Conversely, let  $T$  be a tail of  $XaXaX$ . If  $|T| \leq |XaX|$ , then  $T$  is a tail of  $XaX$  as well, and there is nothing to prove. Suppose that  $|T| > |XaX|$ . Without loss of generality we may assume that  $T$  is a left tail. So,  $XaX$  is a proper prefix of  $T$ . Consider all possible cases depending on kind of the tail  $T$ .

Clearly, that  $T$  is not a non-uniform  $A$ -tail. Actually, since the words  $XaX$  and  $XaXaX$  are  $r_1$ -reduced, the word  $X$  cannot begin with  $aa$ . Now suppose that  $T$  is a non-uniform  $B$ -tail, that is,  $T = (bba)^{2+k}ab$  for some  $k \geq 0$ . Since the word  $XaX$  is a proper prefix of  $T$ , the word  $Xa$  is a prefix of the word  $(bba)^{2+k}$ . Therefore  $X$  has the form  $(bba)^*bb$ . Hence, the words  $XaX$  and  $XaXaX$  have the form  $(bba)^*bb$  as well and have no tails, which is impossible.

If  $T$  is a non-reducible  $B$ -tail, that is,  $T = (bab)^k baba (ba)^l bb$  for some  $k > 0$  and  $l \geq 0$ , then  $X$  begins with  $babb$ , in particular, the word  $X$  is not letter-alternating. On the other hand, the word  $T$  does not contain the factor  $aa$ , therefore  $X[|X|] = b$ . Thus, we have  $baba = X[|X|]aX[1..2] \leq XaX$ . Since  $XaX$  is a prefix of  $T$ , we conclude that  $|Xa| > |(bab)^k|$  and  $X \leq (ba)^{l+2}b$ . So, the word  $X$  appears to be letter-alternating, a contradiction.

Finally, let  $T = (aba)^k abab(ab)^l aa$  for some  $k > 0$  and  $l \geq 0$  ( $T$  is a non-reducible  $A$ -tail). Obviously,  $X[1] = a$ . Since the words  $XaX$  and  $XaXaX$  are  $r_1$ -reduced, we have  $X[|X|] = b$ . Also, from  $aa = aX[1] \leq aX$  it follows that  $|X| < |(aba)^k|$ . Thus, the word  $X$  has the form  $(aba)^*ab$ . It is easy to check that the words  $XaX$  and  $XaXaX$  have the form  $(aba)^*ab$  as well and have no tails. This contradiction completes the proof of statements 4–5.

Now prove statement 6. First, we prove the following assertion:

*if a word  $P$  is a prefix and a suffix of a word  $Q$  and  $Q$  is  $\tilde{A}$ - or  $\tilde{B}$ -whole, then the word  $P$  is  $\tilde{A}$ - or  $\tilde{B}$ -whole respectively as well.*

Indeed, let  $Q$  be  $\tilde{A}$ -whole and  $P = Sa(ab)^k aaT$  for some words  $S, T \in \Sigma^*$ , and integer  $k > 0$ . Since the word  $P = Sa(ab)^k aaT$  is a prefix of the word  $Q$  and  $Q$  is  $\tilde{A}$ -whole, we have  $S = S'ab$  for some  $S' \in \Sigma^*$ . On the other hand, the word  $Sa(ab)^k aaT$  is a suffix of  $Q$ , therefore  $T = baT'$  for some  $T' \in \Sigma^*$ . Thus, we get  $P = S'ab a(ab)^k aa baT'$ , that is, the factor  $a(ab)^k aa$  occurs in  $P$  inside the factor  $(aba)(ab^k)a(aba)$ . This means that the word  $P$  is  $\tilde{A}$ -whole, as desired. The same argument works for a  $\tilde{B}$ -whole word  $Q$ . So, the first part of statement 6 is proved.

To complete the proof of statement 6, suppose that the word  $XaX$  is  $\tilde{A}$ -whole (or  $\tilde{B}$ -whole) and show that the word  $XaXaX$  is  $\tilde{A}$  (resp., or  $\tilde{B}$ -whole) as well. If  $XaXaX$  contains no factor of the form  $a\tilde{A}a$  (resp.,  $b\tilde{B}b$ ), there is nothing to prove. Now let  $XaXaX = RST$ , where  $R, T \in \Sigma^*$ ,

and  $S = a(ab)^k aa$  (resp.,  $S = b(ba)^k bb$ ) for some integer  $k > 0$ . Obviously, either  $S \leq XaX$  or  $aXa \leq S$ . In the first case, the word  $S = a(ab)^k aa$  (resp.,  $S = b(ba)^k bb$ ) occurs in  $XaXaX$  inside the factor  $aba(ab)^k aaba$  (resp.,  $bab(ba)^k bbab$ ), as desired. In the second case, the word  $X$  is letter-alternating. Clearly, if the length of  $X$  is odd, then both words  $XaX$  and  $XaXaX$  are letter-alternating as well, in contradiction with the assumption  $S \leq XaXaX$ . Thus,  $X = (ab)^l$  or  $X = (ba)^l$  for some integer  $l \geq 1$  whence  $XaXaX = (ab)^l a(ab)^l a(ab)^l$  or  $XaXaX = (ba)^l a(ba)^l a(ba)^l$ . Since  $X \notin \{ab, ba\}$  by the lemma's condition, we have  $l > 1$ . Obviously, both words  $(ab)^l a(ab)^l (ab)$  and  $(ba)^l a(ba)^l a(ba)^l$  are  $\widetilde{AB}$ -whole whenever  $l > 1$ . This completes the proof of statement 6.

Now suppose that the words  $XaX$  and  $XaXaX$  are  $\widetilde{AB}$ -whole and have no non-reducible tails. Prove statement 7. By Proposition 1.4, we get  $XaX \sim r(XaX)$ ,  $XaXaX \sim r(XaXaX)$ . Hence, by Proposition 1.3 and Lemma 1.1, the words  $r(X)ar(X)$  and  $r(X)ar(X)ar(X)$  are  $\widetilde{AB}$ -whole, have no non-reducible tails, and satisfy obvious equalities  $r(XaX) = r(r(X)ar(X))$  and  $r(XaXaX) = r(r(X)ar(X)ar(X))$ . So, we will assume that the word  $X$  is already  $r$ -reduced and prove that the words  $XaX$  and  $XaXaX$  are  $r$ -reduced as well.

First, prove that the word  $Xa$  is  $r$ -reduced. If it is not  $r$ -reduced, then  $Xa = X'a(ab)^k aa$  for some  $X' \in \Sigma^*$ . After the reduction  $a(ab)^k aa \rightarrow aa$  inside the words  $XaX$  and  $XaXaX$ , we obtain two equivalent words:

$$\begin{aligned} XaX &= X' \underbrace{a(ab)^k aa}_{\rightarrow aa} X'a(ab)^k a \rightarrow X'aaX'aa(ba)^k = U, \\ XaXaX &= X' \underbrace{a(ab)^k aa}_{\rightarrow aa} \underbrace{X'a(ab)^k aa}_{\rightarrow aa} X'a(ab)^k a \rightarrow X'aaX'aaX'aa(ba)^k = V. \end{aligned}$$

It follows from Proposition 1.4 that  $XaX \sim U$  and  $XaXaX \sim V$ . So, we get  $XaX \sim XaXaX$ , which is impossible. Hence, the word  $Xa$  is  $r$ -reduced.

In a symmetric way one can prove that the word  $aX$  is  $r$ -reduced as well. So, if the word  $XaX$  is not  $r$ -reduced, then it can be decomposed as  $XaX = PcQaRcS$ , where  $P, Q, R, S \in \Sigma^*$ ,  $c \in \Sigma$ ,  $PcQ = RcS = X$ , and the factor  $cQaRc$  has the form  $a\widetilde{A}a$  or  $b\widetilde{B}b$ . Clearly, the prefix  $R$  and the suffix  $Q$  of the word  $X$  do not overlap inside  $X$ , since the letter-alternating word  $R$  cannot contain the factor  $cQ[1] = cc$ . Thus,  $X = RTQ$ , where either  $T = \lambda$  or  $T[1] = T[|T|] = c$ . So, the reduction  $cQaRc \rightarrow cc$  transforms the pair  $(XaX, XaXaX)$  to the pair of neighbours  $(U, V)$ :

$$\begin{aligned} XaX &= RT \underbrace{QaR}_{\rightarrow cc} TQ \rightarrow RTTQ = U, \\ XaXaX &= RT \underbrace{QaR}_{\rightarrow cc} T \underbrace{QaR}_{\rightarrow cc} TQ \rightarrow RTTTQ = V. \end{aligned}$$

Since  $XaX \sim U$  and  $XaXaX \sim V$ , we get  $XaX \sim XaXaX$ , which is impossible. Hence, the word  $XaX$  is  $r$ -reduced.

Finally, suppose that the word  $XaXaX$  is not  $r$ -reduced and let  $S$  be a factor of  $XaXaX$  of the form  $a\widetilde{A}a$  or  $b\widetilde{B}b$ . Then we have  $aXa \leq S$ , because the word  $XaX$  is  $r$ -reduced. So, the word  $X$  is letter-alternating. If the length of  $X$  is odd, then the word  $XaXaX$  is letter-alternating as well, a contradiction with the assumption that the word  $XaXaX$  is not  $r$ -reduced. Hence, the length of  $X$  is even whence  $XaXaX = (ab)^k a(ab)^k a(ab)^k$

or  $XaXaX = (ba)^k a(ba)^k a(ba)^k$  for some  $k \geq 1$ . The condition  $k = 1$  implies that the word  $XaXaX$  is not  $\widetilde{AB}$ -whole. From  $k > 1$  it follows that  $r(XaXaX) = XaX = r(XaX)$  and  $XaXaX \sim XaX$ . Since both case are impossible, we finished the proof of statement 7.

It remains to prove statement 8. Let  $XaX$  and  $XaXaX$  be uniform. Obviously, the word  $X$  is uniform as well. First, suppose that the word  $X$  is letter-alternating. In this case, the length of  $X$  is odd, since the words  $(ab)^k a(ab)^k a(ab)^k$  and  $(ba)^k a(ba)^k a(ba)^k$  are not uniform for any  $k \geq 1$ . Hence, all the words  $XaX$ ,  $XaXaX$ , and  $X$  are odd-length letter-alternating words, therefore  $h(XaX) = h(XaXaX) = h(X) = \lambda$  and  $t(XaX) = t(XaXaX) = t(X) = X[|X|]$ .

Now suppose that  $X[i..i+1] = cc$  for some letter  $c$  and integer  $i$ . Then the factor  $cc$  occurs in  $XaX$  at the  $i$ -th and at the  $|Xa| + i$ -th positions. Since the word  $XaX$  is uniform, we conclude that the length of  $X$  is odd. Hence, all the words  $XaX$ ,  $XaXaX$ , and  $X$  have odd length. Moreover, by the definition of  $\eta$ , if  $i$  is odd, then  $h(XaX) = h(XaXaX) = h(X) = X[1]$  and  $t(XaX) = t(XaXaX) = t(X) = \lambda$ , otherwise  $h(XaX) = h(XaXaX) = h(X) = \lambda$  and  $t(XaX) = t(XaXaX) = t(X) = X[|X|]$ . This completes the proof of the lemma.  $\square$

Call especial attention to pairs of inequivalent words  $(XcX, XcXcX)$  with  $|X| = 2$ . Clearly, the word  $XcX$  is overlap-free in any such pair and  $[XcX]_{r_1} = XcX$ . For the word  $XcXcX$ , we have

$$XcXcX \in \{abaabaab, abbabbab, baabaaba, babbabba, bbabbabb, aabaabaa\} = \mathcal{S}_2.$$

The next lemma describes the equivalence classes of all words from  $\mathcal{S}_2$ .

**Lemma 5.3.**

$$\begin{aligned} [abaabaab]_{r_1} &= (aba)^*(aba)^2ab, & [abbabbab]_{r_1} &= (abb)^*(abb)^2ab, \\ [baabaaba]_{r_1} &= (baa)^*(baa)^2ba, & [babbabba]_{r_1} &= (bab)^*(bab)^2ba, \\ [bbabbabb]_{r_1} &= (bba)^*(bba)^2bb, & [aabaabaa]_{r_1} &= (aab)^*(aab)^2aa. \end{aligned}$$

**Proof.** Clearly, any word of the form  $(aba)^*(aba)^2ab$  is equivalent to  $abaabaab$ . Conversely, if  $W \in [abaabaab]_{r_1}$ , then  $aW \in [(aab)^3]_{r_1} = [aabaab]_{r_1}$ . By Lemma 1.3, we have  $aW = (aab)^k$  for some  $k \geq 2$ . Hence,  $W = (aba)^{k-1}ab$ . Since  $abaab \notin [abaabaab]_{r_1}$ , we have  $k > 2$  and the word  $W$  has the form  $(aba)^*(aba)^2ab$ , as desired. Thus,  $[abaabaab]_{r_1} = (aba)^*(aba)^2ab$ . Now consider the equivalence class  $[bbabbabb]_{r_1}$ . Obviously, this class contains all words of the form  $(bba)^*(bba)^2bb$ . Conversely, let  $W \in [bbabbabb]_{r_1}$ . Then  $Wa \in [bbabba]_{r_1}$  whence  $Wa = (bba)^k$ , where  $k \geq 2$ , by Lemma 1.3. From  $bbabb \notin [bbabbabb]$  it follows that  $k > 2$  and  $W = (bba)^{k-3}(bba)^2bb$ . So, we get  $[bbabbabb]_{r_1} = (bba)^*(bba)^2bb$ . The other equivalence classes are symmetric to described above two classes.  $\square$

Now we ready to prove the main technical lemma of this section.

**Lemma 5.4.** *Let  $(U, V)$  be a bad pair such that the word  $V$  is almost overlap-free. Then there exists a word  $Y \in \Sigma^+$  such that  $V = Y^2$ . Furthermore, denote the word  $Y^3$  by  $W$  and let  $\{Y_k\}_{k=1}^{\ell(Y)}$ ,  $\{V_k\}_{k=1}^{\ell(V)}$ ,  $\{W_k\}_{k=1}^{\ell(W)}$ , and  $\{U_k\}_{k=1}^{\ell(U)}$  be the primary  $Y$ -,  $V$ -,  $W$ -, and  $U$ -series respectively. Then*

- 1)  $\eta(U) \sim \eta(W)$ .
- 2)  $\ell(W) - 1 \leq \ell(Y) \leq \ell(U) = \ell(W) \leq \ell(V) \leq \ell(W) + 1$ .
- 3)  $U_k \sim W_k \not\sim V_k$  for each  $k = 2, \dots, \ell(W)$ .
- 4)  $L_U[k] = L_V[k] = L_W[k] = \lambda$ ,  $R_U[k] = R_V[k] = R_W[k] = \lambda$ ,  $h_U[k] = h_V[k] = h_W[k]$ , and  $t_U[k] = t_V[k] = t_W[k]$  for all  $k < \ell(W)$ ;  
 $L_Y[k] = R_Y[k] = \lambda$ ,  $h_Y[k] = h_W[k]$ , and  $t_Y[k] = t_W[k]$  for all  $k < \ell(Y)$ .
- 5) The words  $Y_k$ ,  $V_k$ , and  $W_k$  are uniform for each  $k = 1, \dots, \ell(W) - 1$ .
- 6) There exists a sequence of letters  $\{c_k\}_{k=2}^{\ell(W)}$  such that  $V_k = Y_k c_k Y_k$  and  $W_k = Y_k c_k Y_k c_k Y_k$  for each  $k = 2, \dots, \ell(W)$  (if  $\ell(Y) = \ell(W) - 1$ , then we put  $Y_{\ell(W)} = \lambda$ ).
- 7)  $Y_{\ell(Y)} \in \{ab, ba, aa, bb, a, b\}$ .

**Proof.** Recall that the words  $U$  and  $V$  are uniform, because the notion of bad pair uses the words  $\eta(U)$  and  $\eta(V)$ . So, the set of all  $r$ -linking  $(V, U)$ -sequences is not empty by Proposition 1.5. For each sequence  $\{R_i\}_{i=0}^n$  from this set, define

$$\beta(\{R_i\}_{i=0}^n) = \text{Card}(\{i \mid (R_{i-1}, R_i) \text{ is bad}, 1 \leq i \leq n\})$$

and

$$\gamma(\{R_i\}_{i=0}^n) = \min\{i \mid (R_{i-1}, R_i) \text{ is bad}\}.$$

Note that the number  $\gamma(\{R_i\}_{i=0}^n)$  is well defined: since the pair  $(V, U)$  is bad, we have  $\beta(\{R_i\}_{i=0}^n) \geq 1$ .

Among all  $r$ -linking  $(V, U)$ -sequences, we choose the sequence  $\{R_i\}_{i=0}^n$  having the lexicographically minimal pair  $(\beta(\{R_i\}_{i=0}^n), \gamma(\{R_i\}_{i=0}^n))$ . The proof consists of four steps. In steps 1–3 we prove that  $\beta(\{R_i\}_{i=0}^n) = 1$  and  $\gamma(\{R_i\}_{i=0}^n) = 1$ , that is, the sequence  $\{R_i\}_{i=0}^n$  has a unique bad pair of neighbours  $(V, R_1)$ . As a by-product of this proof, we will get almost all statements of the lemma. Step 4 is a short argument about the pair  $(R_1, U)$ .

**Step 1.** Let  $\bar{i} = \gamma(\{R_i\}_{i=0}^n)$ . We aim to show that  $\bar{i} = 1$ , i. e.,  $R_{\bar{i}-1} = V$ . To simplify the notation we denote the words  $R_{\bar{i}-1}$  and  $R_{\bar{i}}$  by  $P$  and  $Q$  respectively. By Lemma 5.1, there exists a word  $Y$  such that one of the words  $P$  and  $Q$  is equal to  $YY$  and the other is equal to  $YYY$ . Consider the primary  $Y$ -,  $P$ -,  $Q$ -, and  $V$ -series  $\{Y_k\}_{k=1}^{\ell(Y)}$ ,  $\{P_k\}_{k=1}^{\ell(P)}$ ,  $\{Q_k\}_{k=1}^{\ell(Q)}$ , and  $\{V_k\}_{k=1}^{\ell(V)}$  respectively. Clearly,  $Y_1 = Y$ ,  $P_1 = P$ ,  $Q_1 = Q$ , and  $V_1 = V$ . By the definition of  $P$  and  $\bar{i}$ , the pair  $(V, P)$  is good. Hence, we have  $\eta(V_1) \sim \eta(P_1) \not\sim \eta(Q_1)$  whence  $V_2 \sim P_2 \not\sim Q_2$ . Moreover, we obviously have

$$L_V[1] = L_P[1] = L_Q[1] = L_Y[1] = R_V[1] = R_P[1] = R_Q[1] = R_Y[1] = \lambda$$

and, by Proposition 1.8 and Lemma 5.1,

$$h(V)t(V) = h(P)t(P) = h(Q)t(Q) = h(Y)t(Y) \in \{ab, ba\}.$$

If we denote  $\varphi^{-1}(\eta(Y))$  by  $Y'$ , then we get

$$\varphi^{-1}(\eta(YY)) = Y'\varphi^{-1}(t(Y)h(Y))Y' = Y't(Y)Y'$$

and

$$\varphi^{-1}(\eta(YYY)) = Y'\varphi^{-1}(t(Y)h(Y))Y'\varphi^{-1}(t(Y)h(Y))Y' = Y't(Y)Y't(Y)Y'$$



whence  $\{P_2, Q_2\} = \{Y_2 t(Y) Y_2, Y_2 t(Y) Y_2 t(Y) Y_2\}$  by Lemma 5.2, 1).

Let us prove by induction on  $k$  that

$$\{P_k, Q_k\} = \{Y_k c_k Y_k, Y_k c_k Y_k c_k Y_k\} \text{ and } V_k \sim P_k \not\sim Q_k \quad (5.1)$$

for some sequence of letters  $\{c_k\}_{k=2}^{\ell(Y)}$  and each  $k = 2, \dots, \ell(Y)$ . In addition, we prove that  $|\text{Anc}(Y)| \leq 2$ .

For  $k = 2$ , we put  $c_2 = t(Y)$ , and (5.1) holds, as was shown above. Now suppose that (5.1) holds for some  $k$  such that  $|Y_k| > 2$  and prove that  $k < \ell(Y)$  and (5.1) holds for  $k + 1$  as well.

Obviously, if  $|Y_k| > 2$ , then  $|P_k| > 2$ ,  $|Q_k| > 2$ , and  $|V_k| > 2$  as well. By Lemma 5.2, 2), we have  $V_k \notin \mathcal{S}_1$ . From Lemma 3.1 and Corollary 1.2 it follows that  $k < \ell(V)$ ,  $r_T(P_k) \sim r_T(V_k)$ , and both words  $r_T(V_k)$  and  $r_T(P_k)$  are  $\widetilde{AB}$ -whole.

We claim that  $r_T(P_k) = P_k$  and  $r_T(V_k) = V_k$ , i. e., both words  $P_k$  and  $V_k$  have no non-uniform tails. Assume the converse. Let  $T$  be a tail of the word  $P_k$ . Without loss of generality,  $T$  is a left  $A$ -tail, i. e.,  $T = (aab)^l (aab)^2 ba$  for some  $l \geq 0$ . According to Lemma 5.2, 4), the word  $Q_k$  has the tail  $T$  as well. Since one of the words  $P_k$  and  $Q_k$  equals  $Y_k c_k Y_k$ , we conclude that  $T$  is a prefix of  $Y_k c_k Y_k$ . We have  $c_k = b$ , because  $Y_k$  begins with  $aa$  and the words  $P_k$  and  $Q_k$  are  $r_1$ -reduced. If  $T \leq Y_k$ , then both words  $r_T(Y_k c_k Y_k)$  and  $r_T(Y_k c_k Y_k c_k Y_k)$  have the factor  $aabaabb$  inside the second occurrence of  $Y_k$ . So, these words are not  $\widetilde{AB}$ -whole by definition. But one of them is  $r_T(P_k)$ , which is an  $\widetilde{AB}$ -whole word, a contradiction.

Thus,  $Y_k < T$ . Then  $Y_k = (aab)^{l+2}$  and the words  $r_T(P_k)$  and  $r_T(Q_k)$  have the suffix  $Y_k$ , which is not  $\widetilde{AB}$ -whole. This contradiction proves that  $r_T(P_k) = P_k$ . Since  $P_k \sim V_k$ , by Proposition 1.7, 1) we have  $r_T(V_k) = V_k$ , as desired.

From Lemma 5.2, 4), 6) it follows that all the words  $P_k$ ,  $Q_k$ , and  $Y_k$  are  $\widetilde{AB}$ -whole and have no non-uniform tails. Moreover, since the overlap-free word  $V_k$  has no non-reducible tails, the words  $P_k$ ,  $Q_k$ , and  $Y_k$  have no non-reducible tails as well. Hence, procedure Ancestor cannot stop while processing any of the words  $V_k$ ,  $P_k$ ,  $Q_k$ , and  $Y_k$ , that is,  $k < \min\{\ell(V), \ell(P), \ell(Q), \ell(Y)\}$ .

By Proposition 1.4 and Corollary 1.1, we get

$$r(V_k) = V_k \sim r(P_k) \not\sim r(Q_k).$$

At the same time, we have

$$\{r(P_k), r(Q_k)\} = \{r(Y_k) c_k r(Y_k), r(Y_k) c_k r(Y_k) c_k r(Y_k)\}$$

by Lemma 5.2, 7), 8), where  $r(Y_k)$ ,  $r(P_k)$ , and  $r(Q_k)$  are odd-length uniform words. By definition, the function  $\eta$  deletes exactly one letter from each word  $r(Y_k)$ ,  $r(P_k)$ , and  $r(Q_k)$ . According to Proposition 5.1, we get

$$\eta(V_k) \sim \eta(r(P_k)) \not\sim \eta(r(Q_k))$$

whence  $V_{k+1} \sim P_{k+1} \not\sim Q_{k+1}$ .

Additionally, if we put  $Y'_k = r(Y_k)$ , we get either  $h(Y'_k) \neq \lambda$  and  $t(Y'_k) = \lambda$  or  $t(Y'_k) \neq \lambda$  and  $h(Y'_k) = \lambda$ . Consider the first case. Then

$$\begin{aligned} & \{\varphi^{-1}(\eta(r(P_k))), \varphi^{-1}(\eta(r(Q_k)))\} = \\ & \{\varphi^{-1}(\eta(Y'_k)c_k h(Y'_k)\eta(Y'_k)), \varphi^{-1}(\eta(Y'_k)c_k h(Y'_k)\eta(Y'_k)c_k h(Y'_k)\eta(Y'_k))\}. \end{aligned}$$

Hence,  $h(Y'_k) = \bar{c}_k$  and we get

$$\begin{aligned} & \{\varphi^{-1}(\eta(r(P_k))), \varphi^{-1}(\eta(r(Q_k)))\} = \\ & \{\varphi^{-1}(\eta(Y'_k))c_k\varphi^{-1}(\eta(Y'_k)), \varphi^{-1}(\eta(Y'_k))c_k\varphi^{-1}(\eta(Y'_k))c_k\varphi^{-1}(\eta(Y'_k))\}. \end{aligned}$$

In the second case, we have  $t(Y'_k) = \bar{c}_k$  and we get

$$\begin{aligned} & \{\varphi^{-1}(\eta(r(P_k))), \varphi^{-1}(\eta(r(Q_k)))\} = \\ & \{\varphi^{-1}(\eta(Y'_k))t(Y'_k)\varphi^{-1}(\eta(Y'_k)), \varphi^{-1}(\eta(Y'_k))t(Y'_k)\varphi^{-1}(\eta(Y'_k))t(Y'_k)\varphi^{-1}(\eta(Y'_k))\}. \end{aligned}$$

Thus, if we put  $c_{k+1} = \overline{h(Y'_k)t(Y'_k)}$ , we get

$$\{P_{k+1}, Q_{k+1}\} = \{Y_{k+1}c_{k+1}Y_{k+1}, Y_{k+1}c_{k+1}Y_{k+1}c_{k+1}Y_{k+1}\}$$

in all cases in view of Lemma 5.2, 1).

So, we have proved that

$$\{P_k, Q_k\} = \{Y_k c_k Y_k, Y_k c_k Y_k c_k Y_k\} \text{ and } V_k \sim P_k \not\sim Q_k$$

for each  $k = 2, \dots, \ell(Y)$ , and  $|\text{Anc}(Y)| \leq 2$ . Moreover, we have shown that  $V_k$  is uniform, the words  $V_k, P_k, Q_k$ , and  $Y_k$  have no tails,  $h(V_k) = h(P_k) = h(Q_k) = h(Y_k)$ , and  $t(V_k) = t(P_k) = t(Q_k) = t(Y_k)$  for all  $k < \ell(Y)$ .

Now consider the words  $Y_m = \text{Anc}(Y)$ ,  $Q_m, P_m$ , and  $V_m$ , where  $m = \ell(Y)$ . If  $Y_m \in \{a, b\}$ , then  $P_m$  and  $Q_m$  are odd-length letter-alternating words. Since  $V_m \sim P_m$ , by Observation 1.3 we have

$$V_m = P_m = Y_m c_m Y_m \in \{bab, aba\} \text{ and } Q_m = Y_m c_m Y_m c_m Y_m \in \{babab, ababa\}.$$

Obviously, we have  $\ell(V) = \ell(P) = \ell(Q) = m + 1$ . Note that  $t(Y_m) = Y_m$  in this case. Thus, if we put  $Y_{m+1} = \lambda$ , we get

$$V_{m+1} = P_{m+1} = Y_{m+1}c_{m+1}Y_{m+1} \text{ and } Q_{m+1} = Y_{m+1}c_{m+1}Y_{m+1}c_{m+1}Y_{m+1},$$

where  $c_{m+1} = \overline{h(Y_m)t(Y_m)} = Y_m$ . In the subsequent considerations we put  $Y_{m+1} = \lambda$  if  $Y_m \in \{a, b\}$ .

If  $|Y_m| = 2$ , then the class  $[Y_m c_m Y_m c_m Y_m]_{r_1}$  contains no overlap-free words by Lemma 5.3 and the word  $Y_m c_m Y_m$  is overlap-free. Hence, we have

$$V_m = P_m = Y_m c_m Y_m \text{ and } Q_m = Y_m c_m Y_m c_m Y_m.$$

Clearly, if  $Y_m \in \{aa, bb\}$ , then the words  $V_m, P_m$ , and  $Q_m$  are not  $\widetilde{AB}$ -whole whence  $\ell(V) = \ell(P) = \ell(Q) = m$ . In the case  $Y_m \in \{ab, ba\}$ , we have  $\ell(Q) = m$ , since the word  $Q$  is not  $\widetilde{AB}$ -whole, and  $\ell(V) = \ell(P) = m + 1$ .

So, in all cases we have  $V_m = P_m = Y_m c_m Y_m$ ,  $Q_m = Y_m c_m Y_m c_m Y_m$ , and

$$\ell(Q)-1 \leq \ell(Y) \leq \ell(Q) \leq \ell(P) = \ell(V) \leq \ell(Q)+1.$$

Hence,  $P_k = Y_k c_k Y_k$  and  $Q_k = Y_k c_k Y_k c_k Y_k$  for each  $k = 2, \dots, m$ , and we get  $P = YY$  and  $Q = YYY$ .

Notice that the word  $V_m = P_m$  is uniform if  $\ell(V) = \ell(P) = m + 1$ . It is shown above that the words  $V_k$  are uniform for all  $k < m = \ell(Y)$ . Hence, the words  $V_k$  are uniform for all  $k < \ell(V)$ .

**Step 2.** Consider the direct normal forms of the words  $Y$ ,  $P$ ,  $V$ , and  $Q$  and prove that

$$V = N^D(P) = N^D(Y)N^D(Y) \text{ and } N^D(Q) = N^D(Y)N^D(Y)N^D(Y).$$

Since  $\text{Anc}(P) = \text{Anc}(V)$  is an overlap-free word, the words  $P$  and  $Q$  share the same main direct normal series. Moreover, this series is the primary series of  $V$  by Lemma 4.4.

Let  $\{\tilde{Y}_k\}_{k=1}^{\ell(Y)}$  and  $\{\tilde{Q}_k\}_{k=1}^{\ell(Q)}$  be the direct normal series of the words  $Y$  and  $Q$  respectively. Let  $m = \ell(Q)$  and put  $Y_m = \tilde{Y}_m = \lambda$  if  $m = \ell(Y) + 1$ . Thus, we have

$$V_m = Y_m c_m Y_m = \tilde{Y}_m c_m \tilde{Y}_m \text{ and } \tilde{Q}_m = Q_m = Y_m c_m Y_m c_m Y_m = \tilde{Y}_m c_m \tilde{Y}_m c_m \tilde{Y}_m.$$

Now suppose that

$$V_{k+1} = \tilde{Y}_{k+1} c_{k+1} \tilde{Y}_{k+1} \text{ and } \tilde{Q}_{k+1} = \tilde{Y}_{k+1} c_{k+1} \tilde{Y}_{k+1} c_{k+1} \tilde{Y}_{k+1}$$

for some  $k \in [2; m-1]$ . Prove that the same holds for  $k$ . From Step 1 we know that either  $h_Y[k] = \lambda$  or  $t_Y[k] = \lambda$ . First, assume that  $t_Y[k] = \lambda$ . Then  $c_{k+1} = c_k = \overline{h_Y[k]}$ , and we get

$$\begin{aligned} V_k &= h_V[k] \varphi(V_{k+1}) = h_Y[k] \varphi(\tilde{Y}_{k+1} c_{k+1} \tilde{Y}_{k+1}) = \\ &= h_Y[k] \varphi(\tilde{Y}_{k+1}) c_k \bar{c}_k \varphi(\tilde{Y}_{k+1}) = \tilde{Y}_k c_k \tilde{Y}_k \end{aligned}$$

and

$$\begin{aligned} \tilde{Q}_k &= h_Q[k] \varphi(\tilde{Q}_{k+1}) = h_Y[k] \varphi(\tilde{Y}_{k+1} c_{k+1} \tilde{Y}_{k+1} c_{k+1} \tilde{Y}_{k+1}) = \\ &= h_Y[k] \varphi(\tilde{Y}_{k+1}) c_k \bar{c}_k \varphi(\tilde{Y}_{k+1}) c_k \bar{c}_k \varphi(\tilde{Y}_{k+1}) = \tilde{Y}_k c_k \tilde{Y}_k c_k \tilde{Y}_k, \end{aligned}$$

as desired. Now assume that  $h_Y[k] = \lambda$ . In this case,  $c_{k+1} = t_Y[k] = \bar{c}_k$ , and we get

$$\begin{aligned} V_k &= \varphi(V_{k+1}) t_V[k] = \varphi(\tilde{Y}_{k+1} c_{k+1} \tilde{Y}_{k+1}) t_Y[k] = \\ &= \varphi(\tilde{Y}_{k+1}) \bar{c}_k c_k \varphi(\tilde{Y}_{k+1}) t_Y[k] = \tilde{Y}_k c_k \tilde{Y}_k \end{aligned}$$

and

$$\begin{aligned} \tilde{Q}_k &= \varphi(\tilde{Q}_{k+1}) t_Q[k] = \varphi(\tilde{Y}_{k+1} c_{k+1} \tilde{Y}_{k+1} c_{k+1} \tilde{Y}_{k+1}) t_Y[k] = \\ &= \varphi(\tilde{Y}_{k+1}) \bar{c}_k c_k \varphi(\tilde{Y}_{k+1}) \bar{c}_k c_k \varphi(\tilde{Y}_{k+1}) t_Y[k] = \tilde{Y}_k c_k \tilde{Y}_k c_k \tilde{Y}_k. \end{aligned}$$

So, we have proved that

$$V_k = \tilde{Y}_k c_k \tilde{Y}_k \text{ and } \tilde{Q}_k = \tilde{Y}_k c_k \tilde{Y}_k c_k \tilde{Y}_k$$

for all  $k = 2, \dots, m$ . Finally, consider the words  $V$  and  $\tilde{Q}_1 = N^D(Q)$ . From Step 1 we have  $t_Y[1] = c_2 = \overline{h_Y[1]}$ . Thus, we get

$$\begin{aligned} V &= h_V[1]\varphi(V_2)t_V[1] = h_Y[1]\varphi(\tilde{Y}_2c_2\tilde{Y}_2)t_Y[1] = \\ &h_Y[1]\varphi(\tilde{Y}_2)t_Y[1]h_Y[1]\varphi(\tilde{Y}_2)t_Y[1] = \tilde{Y}_1\tilde{Y}_1 = N^D(Y)N^D(Y) \end{aligned}$$

and

$$\begin{aligned} \tilde{Q}_1 &= h_Q[1]\varphi(\tilde{Q}_2)t_Q[1] = h_Y[1]\varphi(\tilde{Y}_2c_2\tilde{Y}_2c_2\tilde{Y}_2)t_Y[1] = \\ &h_Y[1]\varphi(\tilde{Y}_2)t_Y[1]h_Y[1]\varphi(\tilde{Y}_2)t_Y[1]h_Y[1]\varphi(\tilde{Y}_2)t_Y[1] = \\ &\tilde{Y}_1\tilde{Y}_1\tilde{Y}_1 = N^D(Y)N^D(Y)N^D(Y), \end{aligned}$$

as desired.

Let  $W = N^D(Q)$  and  $Y' = N^D(Y)$ . By Lemma 4.1, we have  $W \sim Q$  and  $W_2 \sim Q_2$ . Let  $\{S_j\}_{j=0}^l$  be a linking  $(W_2, Q_2)$ -sequence. Then the sequence  $\{S'_j = h(Q)\varphi(S_j)t(Q)\}_{j=0}^l$  is an  $r$ -linking  $(W, Q)$ -sequence by Lemma 1.4, 2). From Proposition 1.8 it follows that  $h(S'_j) = h(Q)$  and  $t(S'_j) = t(Q)$  for all  $j \leq l$ . Hence,  $\eta(S'_j) = \varphi(S_j)$  for each  $j = 0, 1, \dots, l$ , and all pairs  $(S'_{j-1}, S'_j)$  for  $j \geq 1$  are good.

Recall that  $(P, Q) = (R_{\bar{i}-1}, R_{\bar{i}})$ , where  $\bar{i} = \gamma(\{R_i\}_{i=0}^n)$ . Suppose that  $\bar{i} > 1$ . Construct a new  $r$ -linking  $(V, U)$ -sequence  $\{R'_j\}_{j=0}^{n'}$ , where  $n' = n + l + 1 - \bar{i}$ , as follows:

$$R'_0 = V, \quad R'_{1+j} = S'_j \text{ for } j = 0, \dots, l, \quad R'_{l+1+i-\bar{i}} = R_i \text{ for } i = \bar{i} + 1, \dots, n.$$

Obviously,  $\beta(\{R'_j\}_{j=0}^{n'}) = \beta(\{R_i\}_{i=0}^n)$  and  $\gamma(\{R'_j\}_{j=0}^{n'}) = 1 < \gamma(\{R_i\}_{i=0}^n)$ , which contradicts the choice of the sequence  $\{R_i\}_{i=0}^n$ . Hence,  $\gamma(\{R_i\}_{i=0}^n) = 1$  whence  $P = V$  and  $Q = R_1$ . Since  $P = YY$  and  $V = Y'Y'$ , we conclude that  $Y = Y' = N^D(Y)$  and  $Q = W$ . So, we have  $V_k = Y_k c_k Y_k$  and  $W_k = Y_k c_k Y_k c_k Y_k$  for all  $k \leq \ell(Y)$ , where  $\{W_k\}_{k=0}^{\ell(W)}$  is the primary  $W$ -series, and  $\tilde{Y}_k = Y_k$  for all  $k \leq \ell(Y)$ .

Moreover, since the word  $V_k$  is uniform for each  $k = 1, \dots, \ell(Y)-1$ , the word  $Y_k$  is uniform for each  $k = 1, \dots, \ell(Y)-1$  as well. In view of Proposition 1.2 and Lemma 5.2, 7), all words  $W_k$  are uniform for  $k < \ell(Y)$  as well. Note that if  $\ell(Y) = \ell(W)-1$ , then  $\text{Anc}(Y) \in \{a, b\}$ . So, the words  $V_k$  and  $W_k$  are uniform for all  $k < \ell(W)$  even if  $\ell(Y) = \ell(W)-1$ .

This completes the proof of statements 5–7 of the lemma. Additionally, we have proved statements 2–4 of the lemma for the words  $V$ ,  $W$ , and  $Y$ . It remains to establish a connection between the words  $W$  and  $U$ .

**Step 3.** On this step, we prove that  $\beta(\{R_i\}_{i=0}^n) = 1$ , i. e., the pair  $(V, W)$  is the only bad pair of neighbours. Assume the converse. Let  $(P, Q) = (R_{\bar{i}-1}, R_{\bar{i}})$ , where  $\bar{i} > 1$ , be the bad pair such that all pairs  $(R_{i-1}, R_i)$  are good for  $1 < i < \bar{i}$ .

By Lemma 5.1, there exists a word  $X$  such that  $\{P, Q\} = \{XXX, XX\}$ . Let  $\{P_k\}_{k=1}^{\ell(P)}$ ,  $\{Q_k\}_{k=1}^{\ell(Q)}$ , and  $\{X_k\}_{k=1}^{\ell(X)}$  be the primary  $P$ -,  $Q$ -, and  $X$ -series respectively. Obviously,  $P_1 = P$ ,  $Q_1 = Q$ , and  $X_1 = X$ . One can easily prove (see Step 1) that  $\{P_2, Q_2\} = \{X_2 d_2 X_2, X_2 d_2 X_2 d_2 X_2\}$ , where  $d_2 = t(X)$ , and

$W_2 \sim P_2 \not\sim Q_2$ . Moreover, we have  $h(X) = h(P) = h(Q) = h(W)$  and  $t(X) = t(P) = t(Q) = t(W)$  whence  $d_2 = c_2$ . Now suppose that

$$\{P_k, Q_k\} = \{X_k c_k X_k, X_k c_k X_k c_k X_k\} \text{ and } W_k \sim P_k \not\sim Q_k \quad (5.2)$$

for some  $k \in [2; \ell(Y)-1]$ , and prove that the same holds for  $k+1$ .

First, we show that  $|X_k| > 2$ . Indeed, if  $|X_k| = 2$ , then the word  $X_k c_k X_k$  is overlap-free,  $[X_k c_k X_k]_{r_1} = X_k c_k X_k$ , and  $X_k c_k X_k c_k X_k \in \mathcal{S}_2$ . Note that all words from  $\mathcal{S}_2$  are not  $\overline{AB}$ -whole. Since the word  $W_k$  is  $\overline{AB}$ -whole, but it is not overlap-free, we get  $W_k \not\sim X_k c_k X_k$  and  $W_k \not\sim X_k c_k X_k c_k X_k$ , contradicting (5.2). In the case  $|X_k| = 1$ , the words  $P_k$  and  $Q_k$  are letter-alternating. Given  $W_k \sim P_k$  and  $V_k \leq W_k$ , we conclude that the words  $W_k$  and  $V_k$  are letter-alternating as well. According to Lemma 5.2, 3), the length of  $W_k$  and  $V_k$  is odd. The inequality  $|Y_k| > 2$  yields  $|V_k|, |W_k| > 5$  whence  $V_k \sim W_k$ , contradicting to the condition that  $(V, W)$  is a bad pair.

So, we have  $|X_k| > 2$  whence  $|P_k| > 2$  and  $|Q_k| > 2$ . Since  $P_k \sim W_k$  and the word  $W_k$  is uniform, Lemma 5.2 implies that both words  $P_k$  and  $Q_k$  are  $\overline{AB}$ -whole and contain neither non-uniform nor non-reducible tails. Moreover, by Proposition 1.4 and Lemma 5.2, 7) we have

$$\{r(P_k), r(Q_k)\} = \{r(X_k) c_k r(X_k), r(X_k) c_k r(X_k) c_k r(X_k)\}$$

and  $W_k \sim r(P_k) \not\sim r(Q_k)$ . Finally, the same argument as in Step 1 gives

$$\{P_{k+1}, Q_{k+1}\} = \{X_{k+1} c_{k+1} X_{k+1}, X_{k+1} c_{k+1} X_{k+1} c_{k+1} X_{k+1}\}$$

and  $W_{k+1} \sim P_{k+1} \not\sim Q_{k+1}$ . Also, we get

$$h(X_k) = h(Q_k) = h(P_k) = h(W_k) \text{ and } t(X_k) = t(Q_k) = t(P_k) = t(W_k).$$

So, by induction on  $k$  we have proved that

$$\{P_k, Q_k\} = \{X_k c_k X_k, X_k c_k X_k c_k X_k\} \text{ and } W_k \sim P_k \not\sim Q_k$$

for each  $k = 2, \dots, \ell(Y)$ . Let  $m = \ell(Y)$ . Consider the words  $X_m$ ,  $P_m$ , and  $Q_m$ . If  $Y_m = aa$  or  $Y_m = bb$ , then the word  $P_m$  has the form  $(aab)^*(aab)^2aa$  or respectively  $(bba)^*(bba)^2bb$  by Lemma 5.3. Clearly,  $X_m = (aab)^l aa$  (resp.,  $(bba)^l bb$ ), where  $l \geq 0$ . If  $l > 0$ , then the words  $P_m$  and  $Q_m$  are equivalent, which is impossible. Hence,  $l = 0$ ,  $X_m = Y_m$ , and  $\{P_m, Q_m\} = \{W_m, V_m\}$ .

In the case  $Y_m = ba$  or  $Y_m = ab$ , the word  $P_m$  has the form  $(abc_m)^*(abc_m)^2ab$  or  $(bac_m)^*(bac_m)^2ba$  respectively. One can easily check that  $X_m = (abc_m)^l ab$  (resp.,  $(bac_m)^l ba$ ), where  $l \geq 0$ . Again, if  $l > 0$ , then  $P_m \sim Q_m$ , which is impossible. Thus, we have  $l = 0$ ,  $X_m = Y_m$ , and  $\{P_m, Q_m\} = \{W_m, V_m\}$ .

Finally, if  $Y_m = a$  or  $Y_m = b$ , then the word  $P_m$  is letter-alternating. By Lemma 5.2, 3), the word  $Q_m$  is letter-alternating as well and the lengths of the words  $X_m$ ,  $P_m$ , and  $Q_m$  are odd. The inequality  $|X_m| \geq 2$  yields  $|P_m|, |Q_m| \geq 5$  whence  $P_m \sim Q_m$ , a contradiction. Hence,  $|X_m| = 1$ , and we get  $X_m = Y_m$  and  $\{P_m, Q_m\} = \{W_m, V_m\}$  again.

So, in all cases we get  $\{P_m, Q_m\} = \{W_m, V_m\}$ . Since  $P_m \sim W_m \not\sim V_m$ , we conclude that  $P_m = W_m$  and  $Q_m = V_m$ . Hence, we have

$$\ell(P) = \ell(W), \quad \ell(Q) = \ell(V), \quad \ell(X) = \ell(Y),$$

and

$$\text{Anc}(P) = \text{Anc}(W), \quad \text{Anc}(Q) = \text{Anc}(V), \quad \text{Anc}(X) = \text{Anc}(Y).$$

Obviously, the primary  $W$ -,  $V$ -, and  $Y$ -series are the direct normal series of the words  $P$ ,  $Q$ , and  $X$  respectively. In particular, we have  $Q_2 \sim V_2$  and  $Q \sim V$ . Let  $\{S_j\}_{j=0}^l$  be a linking  $(V_2, Q_2)$ -sequence. Then the sequence  $\{S'_j = h(V)\varphi(S_j)t(V)\}_{j=0}^l$  appears to be an  $r$ -linking  $(V, Q)$ -sequence such that  $\eta(S'_j) = \varphi(S_j)$  for each  $j = 0, \dots, l$ . Hence, all pairs  $(S'_{j-1}, S'_j)$  for  $j \geq 1$  are good. Now, if we replace the subsequence  $R_0, \dots, R_l$  from the  $r$ -linking  $(V, U)$ -sequence  $\{R_i\}_{i=0}^n$  by the sequence  $\{S'_j\}_{j=0}^l$ , then we get a new  $r$ -linking  $(V, U)$ -sequence with lesser values of  $\beta$  than  $\beta(\{R_i\}_{i=0}^n)$ . We get a contradiction with the choice of  $\{R_i\}_{i=0}^n$ . Hence the pair  $(V, W)$  is a unique bad pair among  $\{(R_{i-1}, R_i) \mid 1 \leq i \leq n\}$ , as desired.

**Step 4.** Since all pairs  $(R_{i-1}, R_i)$  for  $i > 1$  are good, we conclude that the pair  $(W, U)$  is good. In addition, the function  $\eta$  deletes exactly one letter from the word  $W_k$  for each  $k = 1, \dots, \ell(W)-1$ . By Proposition 5.1, condition  $(**)$  from Lemma 3.2 holds for the pair  $(W, U)$ . The rest of the lemma now follows from Lemma 3.2.  $\square$

## 6 Algorithm EqAOF

In this section we complete the construction of Algorithm EqAOF. We start with the description of the second main procedure, called Normalize (procedure Ancestor is introduced in Sect. 3).

### Procedure Normalize.

*Input.* A word  $W \sim \text{Anc}(U)$ , the arrays  $L, R, h, t$  from procedure Ancestor( $U$ ), and the number  $m = \ell(U)$ .

*Output.* A word  $\text{Norm}(U, W)$ .

**Step 1.** If  $m = 1$ , then return  $W$ , stop.

**Step 2.** Let  $m := m - 1$ ;  $W := h[m]\varphi(W)t[m]$ .

**Step 3.** If  $W = Y^3$  for some  $Y \in \Sigma^+$ , then set  $W := Y^2$ .

**Step 4.** Let  $W := L[m]WR[m]$ ; goto step 1.

**End.**

Let  $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2 \cup \{a, b, aa, bb, ab, ba\}$ . We ready to construct Algorithm EqAOF.

### Algorithm EqAOF.

*Input.* An arbitrary word  $U$ .

*Output.* An almost overlap-free word that is equivalent to  $U$  or “FALSE” if no such almost overlap-free word exists.

**Step 1.** Run Ancestor( $U$ ) to get  $\text{Anc}(U)$ , the arrays  $L, R, h, t$ , and the number  $m = \ell(U)$ .

**Step 2.** Find  $W \in \mathcal{S}$  such that  $\text{Anc}(U) \sim W$ ; if no such word  $W$  exists, then return “FALSE” and stop.

**Step 3.** Run Normalize( $W, L, R, h, t, m$ ); Let  $V := \text{Norm}(U, W)$ .

**Step 4.** If  $V$  is almost overlap-free, then return  $V$  else return “FALSE”.

**End.**

The next lemma ensures that Algorithm EqAOF works correctly.

**Lemma 6.1.** *For a word  $U$ ,  $\text{EqAOF}(U) = V$  if there exists an almost overlap-free word  $V \notin \{aaa, bbb\}$  such that  $V \sim U$ , and  $\text{EqAOF}(U) = \text{FALSE}$  otherwise.*

**Proof.** Obviously, if  $\text{EqAOF}(U) = V$ , then  $V$  is an  $r_1$ -reduced almost overlap-free word and  $U \sim V$ . Conversely, suppose that  $U \sim V$ , where  $V$  is an almost overlap-free word such that  $V \notin \{aaa, bbb\}$ . Prove that  $\text{EqAOF}(U) = V$ .

Let  $\{U_k\}_{k=1}^{\ell(U)}$ ,  $\{V_k\}_{k=1}^{\ell(V)}$  be the primary  $U$ - and  $V$ -series respectively, and let  $m = \min\{\ell(U), \ell(V)\}$ . If condition  $(**)$  from Lemma 3.2 holds, then  $\ell(U) = \ell(V)$  and  $U_k \sim V_k$  for all  $k \leq \ell(V)$ . In particular, we get  $\text{Anc}(U) \sim \text{Anc}(V)$ . By Lemma 3.1, 1), we have  $\text{Anc}(V) \in \mathcal{S}$ . Thus, Algorithm EqAOF puts  $W = \text{Anc}(V)$  on Step 2. Since all words  $V_k$  for  $k \leq \ell(V)$  are almost overlap-free, procedure Normalize constructs the main normal  $U$ -series, which coincides with the primary  $V$ -series by Lemma 4.4. So,  $\text{Norm}(U, V) = N(U) = V$ . Hence,  $\text{EqAOF}(U) = V$ , as desired.

Now suppose that there exists an integer  $k' < m$  such that all pairs  $(r(r_T(U_k)), r(r_T(V_k)))$  for  $k < k'$  are good and the pair  $(r(r_T(U_{k'})), r(r_T(V_{k'})))$  is bad. From the proof of Lemma 3.2 it follows that  $U_k \sim V_k$  for all  $k \leq k'$ . Moreover, we get  $L_U[k] = L_V[k]$ ,  $R_U[k] = R_V[k]$ ,  $h_U[k] = h_V[k]$ , and  $t_U[k] = t_V[k]$  for each  $k = 1, \dots, k'$ . Let us denote the words  $r(r_T(U_{k'}))$  and  $r(r_T(V_{k'}))$  by  $U'$  and  $V'$  respectively. By Lemma 5.4, there exists a word  $Y$  such that  $V' = YY$ ,  $\ell(U') = \ell(W')$ , and  $\text{Anc}(U') \sim \text{Anc}(W') \in \mathcal{S}_2$ , where  $W' = YYY$ . Let  $\{W'_k\}_{k=1}^{\ell(W')}$  and  $\{Y_k\}_{k=1}^{\ell(Y)}$  be the primary  $W'$ - and  $Y$ -series respectively. Since  $\text{Anc}(U') = \text{Anc}(U)$ , Algorithm EqAOF, running on  $U$ , chooses  $W = \text{Anc}(W')$  on Step 2.

According to Lemma 5.4, the  $W$ -normal series of  $U'$  coincides with the primary  $W'$ -series. Since  $|W'_m| = |Y_m c_m Y_m c_m Y_m| = 3|Y_m| + 2$ , the word  $W'_m$  is not a cube for each  $m = 2, \dots, \ell(W')$ . Hence, the condition on Step 3 of procedure Normalize is not fulfilled, and procedure Normalize, running on  $W$ , constructs the  $W$ -normal series of  $U'$ , while  $m = \ell(U), \dots, \ell(U) - \ell(U') + 2$ .

On the iteration with  $m = \ell(U) - \ell(U') + 1 = k'$  we obtain the word  $W' = YYY$ . On Step 3 of this iteration, procedure Normalize reduces  $W'$  to the word  $V' = YY$ . Since the word  $V_{k'}$  is almost overlap-free, we get  $V' = r_T(V_{k'})$ . Hence, on Step 4 we obtain the word  $L_U[k']V'R_U[k'] = L_V[k']r_T(V_{k'})R_V[k'] = V_{k'}$ . After that procedure Normalize restores all words  $V_m$  for  $m = k'-1, \dots, 1$  and returns the word  $\text{Norm}(U, W) = V$ . So, Algorithm EqAOF returns  $V$ , as desired.  $\square$

The next lemma estimates the time complexity of Algorithm EqAOF.

**Lemma 6.2.** *Algorithm EqAOF has  $O(|U|)$  time complexity, where  $U$  is an input word.*

**Proof.** Step 1 runs procedure Ancestor. The cycle in Ancestor consists of constant-time and linear-time operations (checks and reductions). The nontrivial check in Step 1 of Ancestor is linear, because two given classes are recognizable languages. So, if the cycle in Ancestor is bounded by  $C|U| + D$  for any given word  $U$ , then the complexity of Ancestor is bounded by  $\sum_{k=1}^{\ell(U)} (C|U_k| + D)$ . Since  $|U_k| \leq 2^{-k+1}|U|$  and  $\ell(U) \leq \log |U|$ , we conclude

that the time complexity of Ancestor is bounded by

$$C|U| \sum_{k=0}^{\infty} \frac{1}{2^k} + D \log |U| = 2C|U| + D \log |U|.$$

Hence, procedure Ancestor runs in  $O(|U|)$  time.

As to the complexity of Step 2 of Algorithm EqAOF, from Lemmas 1.3 and 5.3 it follows that the equivalence class of any word of  $\mathcal{S}$  is a recognizable language. Thus, the word  $\text{Anc}(U)$  is examined by a finite set of fixed finite automata, and the complexity of this step is linear with respect to  $|\text{Anc}(U)|$ . Procedure Normalize, applied on Step 3, runs in  $O(|U|)$  time by the same reason as procedure Ancestor. Finally, a word can be checked for almost overlap-freeness in linear time also. The corresponding algorithm can be constructed, for example, by modifying Algorithm A' from [14]. We see that Algorithm EqAOF has  $O(|U|)$  time complexity.  $\square$

Lemmas 6.1 and 6.2 together prove Theorem 2.

## References

- [1] Bakirov M. F., Sukhanov E. V. *Thue-Morse words and  $\mathcal{D}$ -structure of the free Burnside semigroup*, Proc. Ural State University. Ser. Mathematics and Mechanics, **18**(3) (2000), 5–19 (in Russian).
- [2] Brzozowski J. *Open problems about regular languages*, edited by R.V. Book. Academic Press, New York, Formal Language Theory, Perspectives and Open Problems (1980), 23–47.
- [3] Green J. A., Rees D. *On semigroups in which  $x^r = x$* , Proc. Cambridge Phil. Soc., **48** (1952), 35–40.
- [4] Guba V. S. *The word problem for the relatively free semigroups satisfying  $t^m = t^{m+n}$  with  $m \geq 4$  or  $m \geq 3$ ,  $n = 1$* , Internat. J. Algebra Comput., **3**(2) (1993), 125–140.
- [5] Guba V. S. *The word problem for the relatively free semigroups satisfying  $t^m = t^{m+n}$  with  $m \geq 3$* , Internat. J. Algebra Comput., **3**(3) (1993), 335–348.
- [6] Kaĉourek L., Polák J. *On free semigroups satisfying  $x^r \simeq x$* , Simon Stevin, **64**(1) (1990), 3–19.
- [7] do Lago A. P. *On the Burnside semigroups  $x^n = x^{n+m}$* , Internat. J. Algebra Comput., **6**(2) (1996), 179–227.
- [8] do Lago A. P. *Maximal groups in free Burnside semigroups*, Lect. Notes Comp. Sci. **1380** (1998), 70–81.
- [9] do Lago A. P., Simon I. *Free Burnside Semigroups*, Theoret. Informatics Appl. **35**(6) (2001), 579–595.
- [10] de Luca A., Varricchio S. *On non-counting regular classes*, Theoret. Comput. Sci., **100**(1) (1992), 67–104.



- [11] McCammond J. *The solution to the word problem for the relatively free semigroups satisfying  $t^a = t^{a+b}$  with  $a \geq 6$* , Internat. J. Algebra Comput., **1**(1) (1991), 1–32.
- [12] Plyushchenko A. N., Shur A. M. *Almost overlap-free words and the word problem for Free Burnside semigroup satisfying  $x^2 = x^3$* , Proceedings of WORDS'07, Marseille, France, 245–253
- [13] Plyushchenko A. N. *Overlap-free words and the Free Burnside Semigroup with two generators satisfying  $x^2 = x^3$* . Siberian Electronic Mathematical Reports, **6** (2009), 166–181 (in Russian).
- [14] Shur A. M. *Overlap-free words and Thue-Morse sequences*, Internat. J. Algebra Comput., **6**(3) (1996), 353–367.